AD

SWERR-TR-72-73

COMPUTER OPTIMIZATION OF MACHINING CONDITIONS FOR SHOP PRODUCTION

TECHNICAL REPORT

October 1972

RESEARCH DIRECTORATE
WEAPONS LABORATORY, WECOM
RESEARCH, DEVELOPMENT AND ENGINEERING DIRECTORATE
U. S. ARMY WEAPONS COMMAND

DISPOSITION INSTRUCTIONS:

Destroy this report when it is no longer needed. Do not
return it to the originator.

DISCLAIMER:

The findings of this report are not to be construed as an
official Department of the Army position unless so designated
by other authorized documents.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*	2a. REPORT SECURITY CLASSIFICATION
Pennsylvania State University Department of Industrial Engineering University Park, Pennsylvania	Unclassified
	2b. GROUP

3. REPORT TITLE

Computer Optimization of Machining Conditions for Shop Production

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Inyong Ham

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1972	87	

8a. CONTRACT OR GRANT NO. DAAF01-70-C-1069	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. PRON: A1-9-23045-02-M1-M1	Bulletin B-105
c. AMS Code: 4932.06.6779	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	SWERR-TR-72-73

10. DISTRIBUTION STATEMENT

Approved for public release, distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U. S. Army Weapons Command Research, Development & Engineering Dir. Rock Island, Illinois 61201

13. ABSTRACT

The work described in this report was initiated by personnel of the Research Directorate, Weapons Laboratory, U. S. Army Weapons Command, for application of computer optimization techniques to the study of machining parameters.

Two computer methods for industrial optimization of machining conditions are described and demonstrated. The PERFORMANCE INDEX METHOD (PIM) requires only shop data for machining time, number of pieces produced, and number of tool changes. The PRODUCTION OPTIMIZATION METHOD (POM) requires tool life, time, and cost data. Both are designed to refine the initial data input with shop test data obtained during normal production, as related to one or more of three production objectives: minimum unit cost, maximum production rate and maximum profit rate. The computer programs are constructed for use by shop personnel with little knowledge of mathematics or computers.

Both methods are rapid and economical, and the programs can be processed by either in-plant or remote computer facilities. The user is given all information needed to install the programs and adapt them to his purposes.
(U) (Ham, I.).

DD FORM **1473** REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
UNCLASSIFIED						
1. Machining						
2. Optimization						
3. Computerization						
4. Response surface						
5. Regression analysis						

AD

RESEARCH DIRECTORATE

WEAPONS LABORATORY, WECOM

RESEARCH, DEVELOPMENT AND ENGINEERING DIRECTORATE

U. S. ARMY WEAPONS COMMAND

TECHNICAL REPORT

SWERR-TR-72-73

COMPUTER OPTIMIZATION OF MACHINING
CONDITIONS FOR SHOP PRODUCTION

October 1972

ABSTRACT

The work described in this report was initiated by personnel of the Research Directorate, Weapons Laboratory, U.S. Army Weapons Command, for application of computer optimization techniques to the study of machining parameters.

Two computer methods for industrial optimization of machining conditions are described and demonstrated. The PERFORMANCE INDEX METHOD (PIM) requires only shop data for machining time, number of pieces produced, and number of tool changes. The PRODUCTION OPTIMIZATION METHOD (POM) requires tool life, time, and cost data. Both are designed to refine the initial data input with shop test data obtained during normal production, as related to one or more of three production objectives: minimum unit cost, maximum production rate and maximum profit rate. The computer programs are constructed for use by shop personnel with little knowledge of mathematics or computers.

Both methods are rapid and economical, and the programs can be processed by either in-plant or remote computer facilities. The user is given all information needed to install the programs and adapt them to his purposes.

CONTENTS

Page

CONTENTS (Continued)

FIGURES

PART I. INTRODUCTION

Though computer assisted systems such as numerical control and adaptive control can greatly increase the efficiency of material removal, we still need to improve the economics of this industrial operation. It is the purpose of this report to show how optimization methods employing the computer can reduce costs by finding the most effective machining conditions for a given production objective.

NEW METHODS OF OPTIMIZATION ANALYSIS

Described here are two practical computer methods for analyzing optimum cutting conditions, developed and tested in the Machinability Laboratory, Department of Industrial Engineering, at The Pennsylvania State University. Both are designed to determine from actual production data the best combination of machining parameters in relation to minimum cost per piece, maximum production rate, or maximum profit rate.

The PERFORMANCE INDEX METHOD requires only information on total machining time, number of pieces produced, and number of tool changes, which machine operators can easily supply. It is used when tool life equations, time study data, and cost data are not readily available. This method is based on a concept of response measured directly from production operations (a performance index), by which the optimization criteria are defined. The optimum set of conditions is then found by a mathematical procedure employing production test results from the shop.

The PRODUCTION OPTIMIZATION METHOD requires tool life, time, and cost data generated by production tests. The tool life data are analyzed in relation to other parameters, with repetitive feedback from the shop, until adequate tool life data are obtained. Optimum cutting conditions are determined from the results of this analysis.

Input to the computer programs for these methods is expressed in ordinary machining terms, and the functional engineering decisions are incorporated in the program design. After the basic programs have been constructed, they can therefore be used with varying job conditions by anyone who is capable of supplying the necessary shop data. Both methods are economical of computer time.

ADVANTAGES OF COMPUTER ANALYSIS

The primary advantage of these methods is that they can easily provide information that has heretofore been difficult or impossible to obtain in the manufacturing plant. It is a further advantage that the output of the computer programs is printed instructions and tables, immediately ready for use by plant personnel.

A production engineer who attempts optimization analysis by conventional means has complicated work to do. Though he may have abundant machinability data to draw upon, from sources such as the U.S. Army (14)[*]

---

[*]See Bibliography.

and the U.S. Air Force (13), those data are from experimental investigations and production case studies performed elsewhere. They are not likely to apply exactly to a specific shop job with its own complex variables, and extending them to a particular production case involves many difficulties and precautions. Analyzing tool life data by conventional means requires lengthy calculation of theoretical and empirical relations. Numerous efforts to simplify the practical application of machinability data with worksheets, nomographs, and tables have not achieved much simplification. At best the calculated solutions are approximations, which the engineer has to interpret from his own experience and judgment. Moreover, since optimum conditions vary from job to job, machine to machine, and tool to tool, he has to figure his way through the whole routine repeatedly.

So it is not surprising that in-plant optimization of cutting conditions is not a common practice. Consequently, manufacturers are often unaware that inefficient machining may be a factor obstructing their efforts to reduce the cost of their products on a highly competitive market. And this inefficiency can persist even where processes are computer controlled.

Earlier work by this investigator and others (5,6,7) suggested the value of the computer in analyzing optimum cutting conditions. But the programs then developed were limited to particular applications or were based on generalized machinability ratings or given tool life data. They were of little help in production, and were not comprehensive enough to include some of the interrelated parameters.

The two methods presented here make broad use of the computer's ability to integrate many parameters simultaneously. These programs can calculate rapidly the best combination of machining conditions for any actual cutting job, in terms of the tooling available and the desired production objective. They are designed for effective use by production personnel who know the practical problems and by the machinists who perform the cutting operations. In plants where a digital computer is not yet standard equipment, the programs can be serviced by remote and time-sharing systems.

In addition to its cost-saving potential, computer optimization can be invaluable in analyzing proposed machine work. New materials, new tools, and new machinery present many manufacturing problems that can be solved by this kind of analysis. It can take much of the guesswork out of estimating the physical and economic feasibility of undertaking a job.

COST OF COMPUTER OPTIMIZATION

The following cost data for the demonstrations of the new optimization methods described herein are a conservative indication of the cost of obtaining a complete computer analysis for an actual production case.

Running the computer program for the PERFORMANCE INDEX METHOD required 3.32 seconds of compiler time, 2.17 seconds of execution time, and a total time of 9.0 seconds including input and output times. The input was 847 cards (program and data), and the output was 1285 printed lines. With a time cost of $0.25 per second for 9.0 seconds and a printout rate

2

of $0.001 per line of output, the total cost of the computer analysis was $3.55. This represents the maximum cost of computation. In production, less than one-third of this output would be generated at each iteration, with a corresponding reduction of computer time.

When the computer program for the PRODUCTION OPTIMIZATION METHOD was forced to generate five iterations during one computer run in the simulation mode, the total computer time was 62 seconds and 1486 lines of output were printed. At the given time and printout rates, the cost of the complete analysis was about $17.

Using the information provided in this report, a competent FORTRAN programmer could adapt either of these programs to a particular user's needs in roughly a month's time. Such tailoring would be likely to yield significant savings of computer time, input data, and out records, since the programs are now written to cover a fairly wide range of conditions.

The shop tests called for in both methods can be carried out as normal production machining. Though the cutting conditions to be tested will obviously not be equally efficient, they will all be within a range approaching the optimum.

Savings of production costs as the result of computer optimization will be proportional to the difference between previous shop practice for the analyzed operation and the computed optimum cutting conditions.

VERSATILITY OF THE METHODS

The PERFORMANCE INDEX METHOD can easily be generalized for application to operations other than turning, like milling and drilling, and also for multitool operations. The program described in this report is specifically designed to optimize two parameters, cutting speed and feed, but it can be expanded to add depth of cut or other related factors.

Since the PRODUCTION OPTIMIZATION METHOD depends on analysis of tool life data for a single cutting tool, it could not be extended to a multitool operation without complete revision of the mathematical procedure. Wherever sufficient tool life, cost, and time data can be obtained, this method is recommended for plant use because it requires fewer tests than the performance index method and tends to give more exact results.

PART 2. PERFORMANCE INDEX METHOD

The performance index method (PIM) computes optimum machining conditions for a given production index, from test data limited to machining time, the number of pieces produced per time unit, and the number of tool changes during that time.

PIM is designed to achieve three practical objectives: a) to obtain accurate results with an economical amount of shop testing, b) to provide a flexible system that can accommodate varying job conditions, and c) to make optimization analysis as "automatic" as possible, so that it can be carried out by machine shop personnel.

The PIM computer program automatically performs the mathematical operations indicated in Section 2.2 and makes the necessary computations. Use of the program needs no understanding of the mathematical processes and little knowledge of computers. All of the input and output is in the form of common machining data.

The structure of the program and the simple language code for communicating with the computer are discussed in Section 2.3. Sections 2.4 and 2.5 demonstrate PIM procedure for simulation tests and for actual production analysis.

2.1 CONCEPT AND TECHNIQUES

The performance index method employs the performance index concept, as its name implies, and the general methods of response surface techniques. We shall begin by defining these basic means. Applying them to efficient solution of the problems of optimization analysis required specific adaptive techniques, described in Section 2.2.

PERFORMANCE INDEX CONCEPT

PIM derives its flexibility from the performance index concept (12). A performance index can be any measurable response of the machining operation to changes in the variables or parameters involved (11).

Since the purpose of optimization analysis is to improve the economics of machining, the logical choice of indexes would be unit cost, production rate, or profit rate, alone or in some combination. These are the indexes selected for the PIM program, but any other measurable response can be substituted.

RESPONSE SURFACE TECHNIQUES

A response surface is the surface described by one or more variables and a measured response to those variables. In the PIM program, the response surface is described by the cutting speed, the feed rate, and the chosen performance index. Other variables can be selected, such as depth of cut.

4

The term "response surface techniques" encompasses a large group of methods for exploring response surfaces to find some optimum point (4,12). In general these techniques stem from the assumption that the response surface cannot be described by an exact equation, or that existing data are not complete enough to permit accurate analysis by statistical methods such as regression techniques. The usual method is to search for the approximate optimum point by running tests at several points on the surface.

The aim of every response surface technique is to locate the optimum point in the most efficient manner. But what is efficient for one case may not be efficient for another. A technique employing only a few test points may find an approximation that is acceptable in some applications; in a different problem, an inaccurate guess might reduce the advantage of using so few test points. So it is the wide variety of applications, each with its own characteristics, that explains why so many techniques have been devised.

Unfortunately, none of the existing techniques appear to be simple and flexible enough for use in production. Porter and Summers (12) report methods designed for an adaptive control application. They involve seeking the optimum point by performing a series of single-point tests, analyzing the results, and selecting new test points on the basis of those results. This technique seems quite impractical because of the number of tests needed and the inaccuracy of the final result. The "Box-Wilson" method, described by Duncan (4), takes fewer tests and produces sufficient accuracy, but it apparently requires the services of someone capable of making statistical decisions as the testing progresses. The necessity for keeping a skilled statistician on hand greatly restricts the utility of this method.

PIM also uses response surface techniques to locate the optimum point. The computer program finds both the area containing the optimum point and the precise position of the optimum point. How efficiently it does this work, regardless of the placement of the initial test points on the response surface, is demonstrated by the simulation examples in Section 2.4. Moreover, PIM maintains its efficiency under varying job conditions, and it is capable of optimizing other manufacturing operations in addition to metal cutting.

2.2 MATHEMATICAL PROCEDURE

Mathematical relations for the performance index method are presented in six steps, corresponding to the analytic steps of the PIM computer program. The basic computational technique is the statistical method of least squares regression. Since the functional procedures are written into the program, where they operate automatically, the information in this section will be of interest only to installing engineers and investigators of optimization analysis.

As here described, the PIM program has written into it a subroutine for simulation tests. This subroutine, which was used for the simulation examples in Section 2.4, employs certain mathematical relations that are not called for in the main program. They are discussed at the end of this

5

section. The simulation subroutine can also be useful in preliminary computer studies of real production cases.

STEP 1. SELECTION OF INITIAL TEST POINTS

The first step in analyzing a given response surface is to choose the initial test points. Predetermined tool life data are usually needed to make this choice, but PIM is intended particularly for situations where reliable tool life data are not available. The computer program itself therefore recommends the initial test points. Provision is made, however, for selection of the test points by an experienced engineer when adequate data are at hand.

PIM has two basic functions: a) to find the general area of the optimum point, and b) to pinpoint the actual position of that point. Since much testing can be saved by locating the initial tests in the area where the optimum point is likely to occur, the program is designed to look for expected properties of the response surface.

Certain assumptions are made concerning the shape of the response surface being tested. The main assumption is that the contours of equal values of the performance index selected (cost per piece, production rate, or profit rate) will be parabolic. This means that the problem must be constrained, and that the optimum point will lie on the constraint that closes the open end of the parabola. PIM therefore chooses initial test points in the area of the constraint and the open end of the parabolic contours, as indicated in Fig. 1.

The computer picks nine points, geometrically arranged. For a regression surface with two variables, the number of points should not be less than six. But the regression will give a much more accurate fit if about nine points are supplied, and the regression equation will fit the performance index surface most accurately if the test points are in a geometric pattern.

Figure 1 shows the general scheme for seeking the optimum point. The curves are the expected contours for constant values of the chosen performance index, cost per piece. The constraints are the maximum and minimum speed and feed and the maximum horsepower available for the machining operation (2,8). The computer chooses the nine initial test points so that they fall within the horsepower constraint and span the upper half of the usable feed values and the middle two-fourths of the usable speed values.

As mentioned earlier, the program user can select the nine initial points himself if he has enough reliable data or if he can make a very good educated guess. The points do not have to fall in the area chosen by the computer program. No matter where they are, the PIM program will probably find the optimum point, but convergence to the true optimum may be slower.

STEP 2. COMPUTATION OF THE PERFORMANCE INDEX

We have defined the performance index as any measurable response to the variables in an operation. After the tests at the recommended points

6

PERFORMANCE INDEX CONSTRAINT
(equal cost per piece curves)
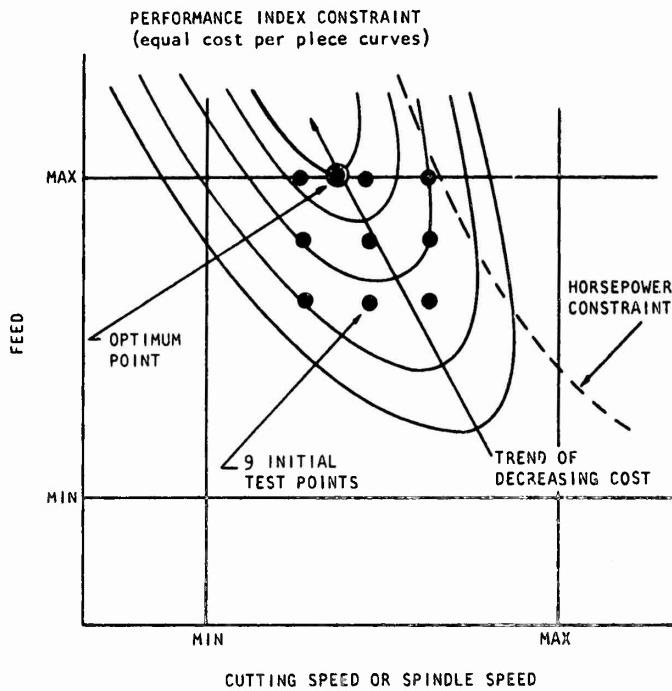
FIG. 1. Scheme for seeking optimum point with
nine initial test points. Usable cutting speed
and feed ranges are restricted by performance
index and horsepower constraints.

have been run, the following information from each test must be fed back
to the computer:

Number of parts produced N_p

Number of tool changes N_{tc}

Time period of test, min T

In the basic PIM program the performance index is a linear combina-
tion of the production rate and the reciprocal of the cost per piece.
But it can be any other measurable response derived from the test results.
The only restriction is that there will be just one optimum point in the
region of testing.

Theoretically, the performance index PI can be defined as

$$PI = QP_r + (1 - Q)(1/C_u) \tag{1}$$

where Q = performance index criterion

P_r = production rate = N_p/T

C_u = cost per piece

The cost per piece or unit cost C_u is determined by

$$C_u = [(RLO)T + (TLC)N_{tc}]/N_p$$

in which   RLO = labor and overhead rate, \$/min

            TLC = tooling cost, \$/edge

The performance index will consist of $Q \times 100\%$ of the production rate and $(1 - Q) \times 100\%$ of the reciprocal of the cost per piece. The value of Q will usually be either 1 or 0, but it can be some intermediate value if a specific percentage ratio between the production rate and the cost per piece is known.

STEP 3. REGRESSION ANALYSIS

When all of the production data for the initial test points have been collected, we are ready to begin analyzing them by the process of least squares regression.

The following set of points is stored by the program:

$$
\begin{array}{ccc}
V_1 & F_1 & PI_1 \\
V_2 & F_2 & PI_2 \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
V_9 & F_9 & PI_9
\end{array}
$$

where   V = cutting speed, fpm, or spindle speed, rpm

       F = feed, ipr

To fit a response surface to these points by multiple regression techniques, we need a generalized equation that will fit the experimental points. If we assume that the surface is well behaved and that there is only one optimum point, the following polynomial equation is a good representation of the response surface within the test area:

$$PI = b_1 + b_2 V + b_3 F + b_4 V^2 + b_5 F^2 + b_6 VF \qquad (2)$$

The error difference e between the experimental value y of PI and the theoretical value of PI given by the right-hand side of Eq. 1 can then be expressed as

$$e = y - (b_1 + b_2 V + b_3 F + b_4 V^2 + b_5 F^2 + b_6 VF) \qquad (3)$$

The purpose of regression analysis is to minimize the sum of the squared error. Since there are actually n such equations, one for each test point, Eq. 3 can be written in matrix form:

$$E = Y - XB \qquad (4)$$

8

in which

$$E = \begin{pmatrix} e_1 \\ e_2 \\ . \\ . \\ . \\ e_n \end{pmatrix} \quad Y = \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{pmatrix} \quad B = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_6 \end{pmatrix} \quad X = \begin{pmatrix} 1 & V_1 & F_1 & V_1^2 & F_2^2 & V_1 F_1 \\ 1 & V_2 & F_2 & V_2^2 & F_2^2 & V_2 F_2 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & V_n & F_n & V_n^2 & F_n^2 & V_n F_n \end{pmatrix}$$

In this form, we can express Σe^2, the criterion to be minimized, as ϕ. Then

$$\phi = (Y - XB)'(Y - XB)$$

$$= Y'Y - 2B'X'Y + B'X'XB$$

and to find the set of b's that will minimize ϕ, the derivative of ϕ is taken with respect to each b_i:

$$\partial\phi/\partial b_i = -2X'Y + 2X'XB = 0$$

$$X'XB = X'Y$$

$$B = (X'X)^{-1}X'Y$$

Given the surface described by Eq. 2, it is now possible to find the optimum point on that surface.

STEP 4. OPTIMIZATION OF PERFORMANCE INDEX

The optimum point could be located by a mathematical optimizing method. But since we are dealing with discrete points and a high speed computer, the most efficient way to locate the desired point is a simple exhaustive search. The optimum point estimated by the regression surface is easily found by calculating, with Eq. 2, the value of PI at every possible combination of feed and speed within the known constraints.

Because the theoretical regression surface as defined by Eq. 2 has only one optimum point, that is, only one maximum or minimum, the search for the optimum can extend beyond the region of the tested points. If the optimum point lies within the region of testing, the regression equation will estimate its position fairly accurately. If it lies outside the tested area, the regression equation will give a general estimate of the position or will at least show the direction to look for further test points.

STEP 5. FURTHER TESTS FOR OPTIMIZATION

After an initial estimate of the optimum point has been obtained, further testing is necessary either to confirm that point or to find a new estimate. The position of the estimated point will determine what values the program will recommend for the next set of tests. The location of these test points on the response surface will depend on whether or not the estimated optimum point is on a constraint.

9

Optimum Point on a Constraint. As stated before, we expect that the optimum point will be found on a constraint. The program is therefore structured to look for this probability and take advantage of it.

When the estimated optimum point occurs on one of the upper constraints (maximum feed or maximum horsepower), further testing is done along that constraint. A curvilinear regression model is used to fit the test points to the constraint, with reference again to the general equation for the surface (Eq. 2):

$$PI = b_1 + B_2 V + b_3 F + b_4 V^2 + b_5 F^2 + b_6 VF$$

Where $F = F_{max}$,

$$PI = b_1 + b_2 V + b_3 F_{max} + b_4 V^2 + b_5 F^2_{max} + b_6 VF_{max} \tag{5}$$

We can group the constants and redefine them as

$$b'_1 = b_1 + b_3 F_{max} + b_5 F^2_{max}$$

$$b'_2 = b_2 + b_6 F_{max}$$

$$b'_3 = b_4$$

The result is a new regression equation:

$$PI = b'_1 + b'_2 V + b'_3 V^2 \tag{6}$$

If there is a horsepower constraint, the maximum horsepower HP_m is defined by

$$HP_m = [(\pi D) FV (DC) S_{hp}]/(eff) \tag{7}$$

where D = diameter of workpiece, in.

 V = spindle speed, rpm

 DC = depth of cut, in.

 S_{hp} = specific horsepower, hp/in.^{3}/min

 eff = efficiency of machine tool, %

We can rewrite Eq. 7, grouping the constants in a new constant C:

$$C = (HP_m)(eff)/[\pi D(DC)S_{hp}] = FV \quad \text{or} \quad C = FV \tag{8}$$

Solving for feed,

$$F = C/V$$

and substituting the F into Eq. 2 results in

$$PI = b_1 + b_2 V + b_3 (C/V) + b_4 V^2 + b_5 (C/V)^2 + b_6 C \qquad (9)$$

which can be rearranged as

$$PI = b_1'' + b_2'' V + b_3'' (1/V) + b_4'' V^2 + b_5'' (1/V^2) \qquad (10)$$

If the values of V are fairly large, say in the hundreds or more, the terms with V or V^2 in the denominator will be significantly small and can be dropped from the equation, which then becomes

$$PI = b_1''' + b_2''' V + b_3''' V^2 \qquad (11)$$

Equations 6 and 11, since they are of the same form, can be joined as one continuous regression model for the purpose of this program. Note, however, that Eq. 11 is the projection of the intersection of the horsepower plane and the PI surface, on the plane formed by V and PI coordinates. It is not the actual curve, but the *projection* of the curve on a plane. As such, it is usable in the program because the projection clearly illustrates the maximum value of PI that is being sought.

The test points must be taken along the constraints in question, but the tests will be fitted to the regression model given by Eq. 6. The constraints that will be tested for the optimum value of PI are shown in Fig. 2. The circled black dot indicates the estimated position of the optimum point, and the open circles indicate the approximate position of the points recommended by the program for further testing.

Optimum Point Not on a Constraint. In this case, one of two things may have occurred. Either the final position of the optimum has not yet been pinpointed, or the performance index surface is not as predicted. If either of these conditions exists, the program will recommend test points closely grouped around the previous estimate, and the surface regression model (Eq. 2) will again be used to fit these new tests to the surface.

STEP 6. STOPPING CRITERION AND ERROR ANALYSIS

The stopping criterion for the PIM program is that whenever the optimizing search produces the same point twice in succession, the program stops. It is based on the opinion that if two sets of test points yield the same result, that result can be accepted with reasonable safety. Because the program is dealing with discrete points, this restriction is not so tight as it may seem. When the computer's exhaustive search locates an optimum point, the actual optimum probably does lie somewhere within the area close to the computer's recommendation.

How tight the stopping criterion really is depends on the distance between the related speed and feed settings. Figure 3 illustrates this situation. The open circles indicate a few of the possible feed and speed settings, and the circled black dot indicates the recommended optimum

11

point. The shaded block represents the area that can be assumed to contain the true optimum.

Steps 1 through 5 are repeated until the stopping criterion is met. The resulting optimum conditions can be taken to represent the best estimate of the true optimum, within the limits of experimental error.

Definite problems will arise if the experimental or production test data contain a large amount of error or variation. During production tests, every effort should be made to reduce the effects of operating variables such as the operator's speed and time out for repairs. Large errors will distort the shape of the performance index surface, produce variations in the optimum point from one run to the next, and disrupt the stability of the analytic method.

The amount of error may be so great that the program will not find an optimum point according to the stopping criterion, but will oscillate around the point. If the area of oscillation is small, the user can decide to pick the optimum at any point within it. With so much error in the input data, an optimum point located by the computer would not be very accurate anyway. To obtain the accuracy of which PIM is capable, errors in the test data should be minimal.

Statistical error might also affect the stopping criterion. This kind of error relates to how well the regression surface fits the actual performance index surface. Least squares regression can minimize statistical error, but cannot completely eliminate it. If the area of testing is small enough so that the enclosed surface is smooth and well behaved and has only one optimum point, the fit of the regression surface should not introduce any significant error.

COMPUTER SIMULATION TESTS

The validity and flexibility of PIM were proved by the computer simulation tests described in Section 2.4. These tests required a subroutine for simulation of the machining data (MACSIM), to evaluate the performance index. Note that this subroutine is used only in simulation tests. It is not a part of the PIM main program for production application, which is independent of predetermined tool life, cost, and time study data.

To compute the performance index, as previously defined for an actual run, we start with a known tool life equation and a set of suitable cost data and time study data. From this information, the production rate and unit cost can easily be calculated by the following mathematical procedure.

Let us take as the tool life equation

$$VT^{\alpha}F^{\beta} = C$$

where V = cutting speed, fpm α, β = exponents
       T = tool life, min   C = constant
       F = feed, ipr

PERFORMANCE INDEX CONSTRAINT
(equal cost per piece curves)
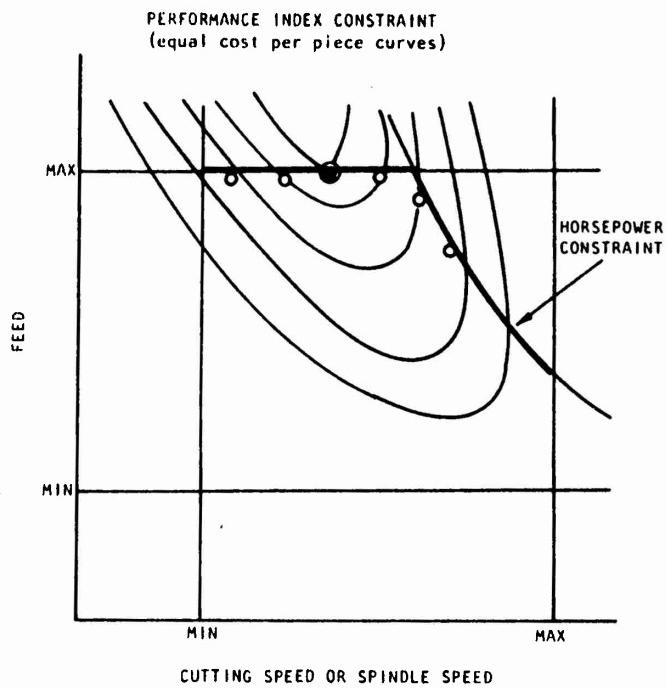
FIG. 2. Optimum point on upper constraint.
Open circles are points recommended by com-
puter program for further testing.
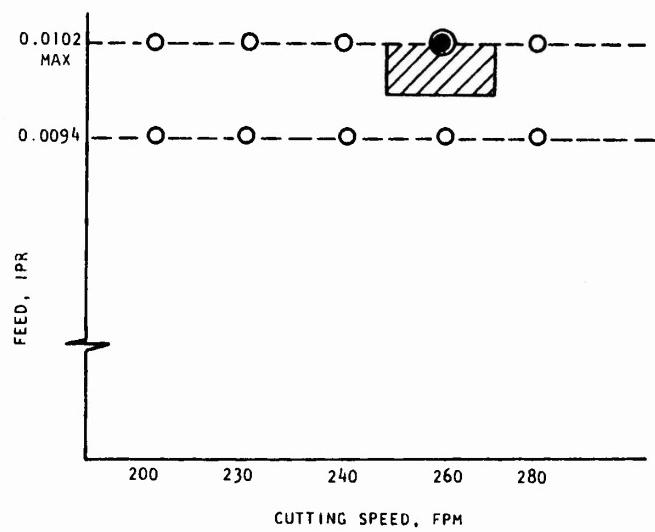
FIG. 3. Final value of discrete optimum
point, and area of confidence for true op-
timum point.

13

and where

$$V = \pi DN/12 \quad \text{or} \quad N = 12V/\pi D$$

in which   D = diameter of workpiece, in.
           N = spindle speed, rpm

We first solve for tool life T:

$$T = (C/VF^\beta)^{1/\alpha}$$

We next calculate T_c, machining time per piece:

$$T_c = L/NF$$

where L = length of workpiece, in.

We can now calculate the total cost per piece C_u:

$$C_u = C_o T_m + C_o T_c + (C_o T_{ct} T_c)/T + C_e T_c/T$$

where C_o = labor rate plus overhead, \$/min
        T_m = machine handling time, min
        T_{ct} = tool changing time, min
        C_e = tool cost, \$/edge

And we can also calculate P_r, the production rate:

$$P_r = 1/[T_m + T_c + T_{ct}(T_c/T)]$$

The performance index PI can then be computed as defined in Eq. 1:

$$PI = QP_r + (1 - Q)(1/C_u)$$

To lend realism to the computer simulation tests, a "random percentage error" was added to each value of PI returned to the subroutine, by writing into the simulation program an error noise of 5% or 10%.

2.3 COMPUTER PROGRAM FOR PIM

The computer program for the performance index method implements the mathematical procedure explained in Section 2.2. It employs the language of FORTRAN IV and can be translated with any FORTRAN compiler. The computer used in the developmental work was the IBM 360, Model 67, in the Computation Center at The Pennsylvania State University. Since the program does not need a large amount of storage, it can be modified to fit many other computer systems. It can also be adapted to optimize other

14

machining operations, such as milling and drilling, simply by changing the input and output formats.

PIM is designed to provide maximum flexibility, at the same time ensuring efficient and accurate performance. As in any other analytic process involving variables and testing, there is a possibility that the program might not find an optimum point in a reasonable time. But because of the nature of the optimization problem and the analytic techniques, the probability of failure should be minimal within the practical limits of testing.

This section describes decisions written into the program, several options available to the user, the coding of input data, and finally the structure of the PIM computer program.

DECISIONS AND OPTIONS

The PIM optimization procedure requires certain engineering decisions. To make the program usable by shop personnel who may be unfamiliar with the mathematical method and may know little about computers, these decisions are written into it and will operate automatically unless they are changed. For flexibility, several options can be exercised to suit specific applications.

Distance Between Adjacent Test Points. Though the analysis can accommodate either a stepped or a stepless machine tool, a step-size factor is required for both cases. All step sizes are expressed in terms of a possible number of speed and feed settings, not as differences between the actual values of the variables. That is, the distance between any two points is some number of settings, usually a fraction of the total number available. This fraction was set at 1/4 of the total number, as the result of a series of tests that proved it to ensure the greatest degree of generality in the use of the program. By this engineering decision, the program automatically computes step sizes as 1/4 of the possible number of step settings, either actual or assumed.

Constraints. Speed, feed, and horsepower constraints are written into the program to ensure that the recommended optimum condition will always be within the capability of the machine tool to be used. Though an optimization analysis can be performed without any constraints, the results will be unrealistic. When values for specific horsepower and machine efficiency are unobtainable, the PIM program can be run without the horsepower constraint. On the other hand, it is possible to read in an additional constraint such as surface finish, if necessary, to further restrict the usable feeds and cutting speeds (9).

Initial Test Points. PIM is programmed to recommend the initial test points. But if the necessary data are available, the user can bypass this part of the program and pick the initial test points himself (Section 2.5, Selection of Initial Test Points). He must bear in mind however, that at least six points should be chosen, and that they should be arranged in a closely grouped geometric pattern to secure the best regression fit (Section 2.2, Step 1).

15

Substitution of Test Points. The user also has the option of replacing any test point recommended by the computer with a point of his own choosing. When the recommended points are all on the upper constraints, the new point must also be on an upper constraint. Any replacement must be made with care not to destroy the mathematical procedure and the general trend of the method. Instead of a substitution, any one or two points may simply be dropped, as long as the remaining points are not fewer than the minimum number needed for the regression (six points for surface regression, three points for curvilinear regression).

Choice of Spindle Speed or Cutting Speed. As the program is now written, the two parameters affecting the performance index are spindle speed N, rpm, and feed F, ipr. Spindle speed was chosen, rather than the cutting speed V, fpm, commonly used in analyses of machining operations, because shop personnel are often more inclined to think in terms of it. Spindle speed can easily be converted to the corresponding cutting speed, and either is acceptable in the program. But if cutting speed is chosen, the horsepower equation in Step 5 of the mathematical procedure (Section 2.2) will have to be rewritten accordingly, and the computer must be instructed to print out cutting speed, fpm, instead of spindle speed, rpm, as it now does. The user has also the option of substituting some other variable, like depth of cut, for either of the two parameters.

Number of Parameters. The mathematical procedure can handle any two discrete parameters. In fact, it is adaptable to any number of parameters, and to any performance index that has only one maximum value in the test area. For more than two parameters the program would have to be suitably adjusted, of course. But even three parameters will create practical problems, because the number of test points needed for regression increases greatly with each additional parameter.

INPUT DATA CODE

The following tabulation lists and defines the code used for input data in the PIM program. For convenience, the corresponding mathematical notation is also shown.

Input Code	Notation	Definition
IDENT		Type of machine tool: = 1 if speed is stepless. Computer calculates artificial steps. = 0 if speed is stepped. Steps are read in on data cards.
VLOW	V_1	Lowest usable spindle speed, rpm.
VFLOW	V_1	Lowest usable cutting speed, fpm, corresponding to V_1.
VHIGH	V_n	Highest usable spindle speed, rpm.
VFHIGH	V_n	Highest usable cutting speed, fpm, corresponding to V_n.

16

Input Code	Notation	Definition
FLOW	F_1	Lowest usable feed, ipr.
FHIGH	F_n	Highest usable feed, ipr.

[Value of FHIGH must be the value of a feed setting. If the desired maximum feed is about 0.0100 ipr, the closest feed setting must be chosen (say, 0.0102 ipr). Otherwise program will not execute properly. If FHIGH is left blank or read in as zero, program will set as FHIGH the highest feed read in. Similarly, if zero values are read in for FLOW, VLOW, or VHIGH, computer will set lowest or highest value read in.]

Input Code	Notation	Definition
SHP	S_{hp}	Specific horsepower, hp/in.^{3}/min.
HPM	HP_m	Maximum horsepower.
EFF	eff	Efficiency of machine tool, percentage in decimal form (for example, 0.5).
DC	DC	Maximum depth of cut expected, in.
RLO	RLO	Labor and overhead rate, $/min.
TLC	TLC	Tool cost, $/edge.
Q	Q	Performance index criterion. Usually 0 or 1, but may be some value between.
D	D	Diameter of workpiece, in.

[The following three variables apply only if IDENT = 1.]

Input Code	Notation	Definition
VI	V_i	Lowest available spindle speed, rpm.
VA	V_a	Highest available spindle speed, rpm.
VCH	V_{ch}	Desired size of steps between V_i and V_a.

[The following two variables apply only if IDENT = 0.]

Input Code	Notation	Definition
L	ℓ	Number of spindle speed settings.
V(I)	V_i	Values of spindle settings, rpm, i = 1 to ℓ. Must be in ascending order, V_1 lowest.
M	M	Number of feed settings.
F(I)	F_i	Values of feed settings, ipr, i = 1 to M. Must be in ascending order, F_1 lowest.
KT		Sequence number for successive computer runs. KT = 0 for initial run.

(CONTINUED)

Input Code	Notation	Definition
IR		Code for type of regression and use of test results. Value for IR is found on print of previous run. IR = 0 for initial run.
NPT or IF		Number of test results to be read in. NPT and IF are used interchangeably. Value is found on print of previous run. If not all test points recommended were run, value should be actual number of tests. IF = 0 for initial run.
VMAX or NOP	V_{max}	Previously calculated value of optimum spindle speed, found on print of previous run. NOP = 0 for initial run.
FMAX or FOP	F_{max}	Previously calculated value of optimum feed, found on print of previous run. FOP = 0 for initial run.

[The following five variables are read in only if NPT $\neq$ 0.]

VL(I)	V_i	Value of spindle speed used in a test run.
FD(I)	F_i	Value of feed used in a test run.
P(I)	N_{pi}	Number of parts produced in time T at test values of V_i and F_i.
T(I)	T_i	Time period of test run, min.
TP(I)	N_{tci}	Number of tool changes in time T at test values of V_i and F_i.

COMPUTER PROGRAM

The PIM computer program is described completely in Appendix A, which gives the flow diagram, a format guide for data input cards, and the detailed program listing. As shown, both the flow diagram and the program listing contain some elements that apply only to the simulations in Section 2.4. These are indicated by broken lines in the flow diagram, and by C* with verbal instructions in the program listing.

The format guide should enable a person who has only slight acquaintance with computers to feed data into this program. Not all of the 10 types of data cards are used for any one run, of course. When the original data is submitted, card 9 is included but left blank (zero values). For successive runs, card 9 will contain KT (sequence number) and the values of IR, IF, NOP, and FOP supplied by the computer.

The form of the program's output is described in the next two sections, and is shown in Appendix A as the entire printout for the production example.

2.4 COMPUTER SIMULATION EXAMPLES

The capabilities of the performance index method were demonstrated by computing five simulation examples with the PIM program. The same hypothetical data were used for all of the examples, but each displayed a variation of procedure or of some operating condition.

The detailed description of Example 1 indicates the general patterns of input and output for all of the simulation cases. The shape of each optimization analysis and its results are shown graphically in Figs. 4 to 9.

ASSUMED DATA

Operation and Tooling. The hypothetical job, tooling, and material for the simulation examples were taken to be

Operation	Straight turning
Machine tool	Engine lathe (IDENT = 1)
Workpiece	Custom-455 steel, 6 in. diam, 24 in. long

Performance Index. The same performance index was used for all of the analyses -- cost per piece, Q = 0. To evaluate this index by means of the subroutine described under COMPUTER SIMULATION TESTS in Section 2.2, the following data were assumed:

Time data	Tool changing time	1 min
	Machine handling time	5 min
Cost data	Labor and overhead rate	$0.10/min
	Cutting tool cost	$0.20/edge
Machine data	Maximum horsepower	7.5
	Specific horsepower	$0.75/in.^3/min$
	Machine efficiency	60%

Tool life data $\quad VT^{0.203}F^{0.194} = 207$

Constraints. In addition to the constraints imposed by horsepower and machine efficiency, the machine tool had assumed limits of capability:

Available spindle speed range	20 to 1000 rpm in 50 steps
Available feed range	0.0011 to 0.0168 ipr in 24 steps

The given machining operation imposed further restrictions:

Usable speed range	200 to 1000 fpm in 26 steps (127 to 637 rpm)
Usable feed range	0.0051 to 0.0102 ipr in 9 steps

SIMULATION EXAMPLE 1

This is a simple case in which the optimum condition is found by only two steps of optimization analysis. As indicated by the other examples, more steps are usually required. The analysis of this example is illustrated in Fig. 4.

19

The variants for Example 1 were as follows:

Initial test points   Selected by computer
Feed range   Roughing condition, 0.0051 to 0.0102 ipr
Error added   5%

In all of these simulations, a cost per piece function was used to represent actual production costs. To better approximate shop results, an arbitrary error percentage, usually 5% but in one case 10%, was added to the computed costs. That is, the indicated percentage of the cost was taken and multiplied by a random number that was generated from -1 to +1, and this random error was then added to or subtracted from the cost calculated by the cost per piece function.

Data Input. Since the computer is going to determine the initial tests points, we need the following data input to the program. The code terms are as defined in Section 2.3, and the numerical data are from our general assumptions for the simulations and the variant conditions given for Example 1.

IDENT	= 1	M	= 24	
VLOW	= 127.0	F(I)	= 0.0011	
VHIGH	= 637.0		to	24 steps
			0.0168	
FLOW	= 0.0051	KT	= 0	
FHIGH	= 0.0102	IR	= 0	
VFLOW	= 200.0	NPT	= 0	One blank card for these five zero values
VFHIGH	= 1000.0	VMAX	= 0.0	
SHP	= 0.75	FMAX	= 0.0	
HPM	= 7.5	VL(I)		
EFF	= 0.60	FD(I)		
DC	= 0.1	T(I)		Required only when NPT $\neq$ 0, as in Example 3
RLO	= 0.1 [C_o]	P(I)		
TLC	= 0.2 [C_e]	TP(I)		
Q	= 0.			
D	= 6.			
VI	= 20.0			
VA	= 1000.0			
VCH	= 20.0			

The information for evaluating the performance index in the simulation program is supplied by a data statement in the subroutine MACSIM. The statement reads:

DATA A,BB,C,TCT,CO,CE,TM,D,XL/0.203,0.194,270.,1.,0.1,0.2,5.,6.,24./
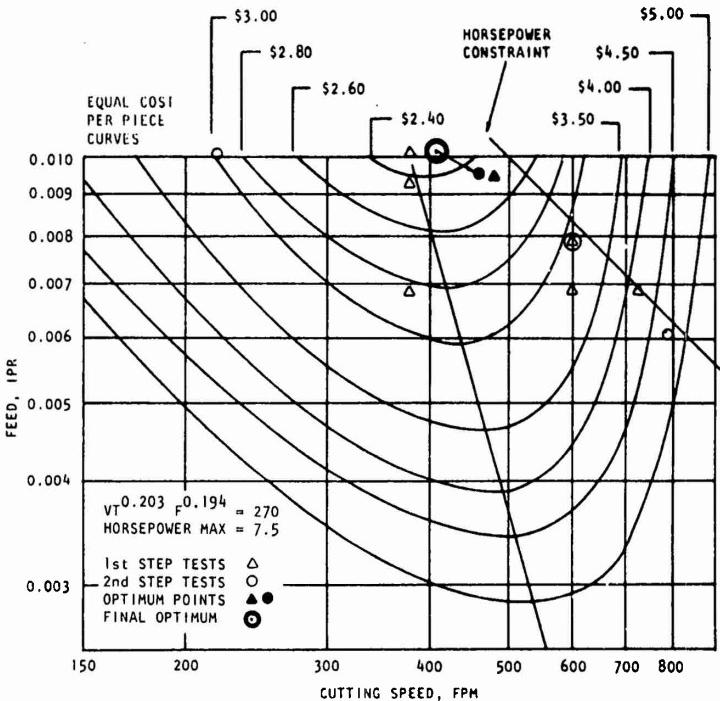
FIG. 4. Computer simulation Example 1, optimization
analysis. Cost per piece index with 5% error, initial
test points selected by computer, roughing condition.

in which

A	= alpha, exponent of tool life	=	0.203
BB	= beta, exponent of feed	=	0.194
C	= constant in tool life equation	=	270.
TCT	= tool changing time	=	1.
CO	= labor and overhead rate	=	0.1
CE	= tool cost per edge	=	0.2
TM	= machining time	=	5.
D	= diameter of workpiece	=	6.
XL	= length of workpiece	=	24.

The error percentage is set by a card in the subroutine as: ERR = 0.05.

Initial Test Points. The computer selected six initial test points
for Example 1:

Cutting Speed		Feed,
rpm	fpm	ipr
240	377	0.0068
240	377	0.0102
240	377	0.0092
380	597	0.0068
380	597	0.0078
460	723	0.0068

21

The location of these points is shown by the open triangles in Fig. 4. The program is expected to choose points in a geometric arrangement, but in this case the choice was influenced by the horsepower constraint.

First Optimization. The printout for the first optimization of the simulated data read:

| Cutting Speed | | Feed, | Unit Cost | Prod. Rate |
rpm	fpm	ipr	Index, CU	Index, PR
240	377	0.0068	2.169	0.05
240	377	0.0092	1.813	0.06
240	377	0.0102	1.697	0.07
380	597	0.0068	2.611	0.06
380	597	0.0078	2.371	0.06
460	723	0.0068	3.887	0.05

Based on these data, the program recommended the following cutting condition as the optimum conditions to be tested further:

Cutting speed = 260 rpm or 408.4 fpm

Feed = 0.0102 ipr

Further Optimization. To refine the results obtained with the initial test points, the computer then recommended four more points for testing:

| Cutting Speed | | Feed, |
rpm	fpm	ipr
140	220	0.0102
260	408	0.0102
380	597	0.0078
500	785	0.0060

The simulation program's analysis of these points gave the following data:

| Cutting Speed | | Feed, | Unit Cost | Prod. Rate |
rpm	fpm	ipr	Index, CU	Index, PR
140	220	0.0102	2.223	0.04
260	408	0.0102	1.599	0.07
380	597	0.0078	2.509	0.06
500	785	0.0060	4.718	0.04

For these data, the recommended optimum conditions were again

Cutting speed = 260 rpm or 408.4 fpm

Feed = 0.0102 ipr

Stopping Criterion. Since the optimum conditions proved to be the same for the second run as for the first, the stopping criterion had been met and no further testing was indicated. For the given machining operation, the heavily circled point in Fig. 4 is the final optimum.

SIMULATION EXAMPLE 2

The only difference between this example and Example 1 is that the error level was raised to 10%. Because the increased error caused more oscillation of the optimum points, finding the final optimum required three steps of analysis. Nevertheless, the result for Example 2, shown in Fig. 5, was the same as that for Example 1.

SIMULATION EXAMPLE 3

In all other respects the same as Example 1, this simulation demonstrates selection of the initial test points by the user.

For Example 3, nine initial points were chosen at an arbitrary location within the given limits of the constraints. To rule out computer selection, the following data were added to the original input:

NPT = 9

VL(I) = 3 values ⎫
                  ⎬ 9 cards
FD(I) = 3 values ⎭

The cutting speeds chosen for this simulation were 140, 150, and 160 fpm. The feeds were 0.0051, 0.0060, and 0.0078 ipr. For each VL(I) there were three cards combining that value with each of the three FD(I) values. With these nine cards in the deck, the computer program was ready to begin the optimization procedure.

Because of some degree of oscillation of the optimum points in the intermediate steps, reaching the final optimum took five steps of optimization analysis. As shown in Fig. 6, the PIM program can eventually find the optimum point regardless of how the initial test points were chosen. But manual selection is likely to take more test steps.

SIMULATION EXAMPLE 4

To test the versatility of the PIM program, in Example 4 the initial test points, again manually chosen, were located far from the expected optimum points, and the feed range was broadened to 0.003 to 0.0102 ipr, which would cover both roughing and finishing conditions.

Figure 7 shows that the program's analysis of this simulation arrived at the same recommended optimum conditions as in all previous cases, 260 rpm or 408.4 fpm and 0.0102 ipr. It shows also that five steps of analysis were needed to get there. Although the optimum point can be found with any set of initial test points, the closer the test points are to the expected optimum points, the shorter the analysis.

SIMULATION EXAMPLE 5

For this example we return to nine initial test points selected by the computer program. The purpose of Example 5 is to demonstrate how the program works with a different machining constraint, the feed range reduced to 0.003 to 0.0051 ipr for finishing conditions.

23
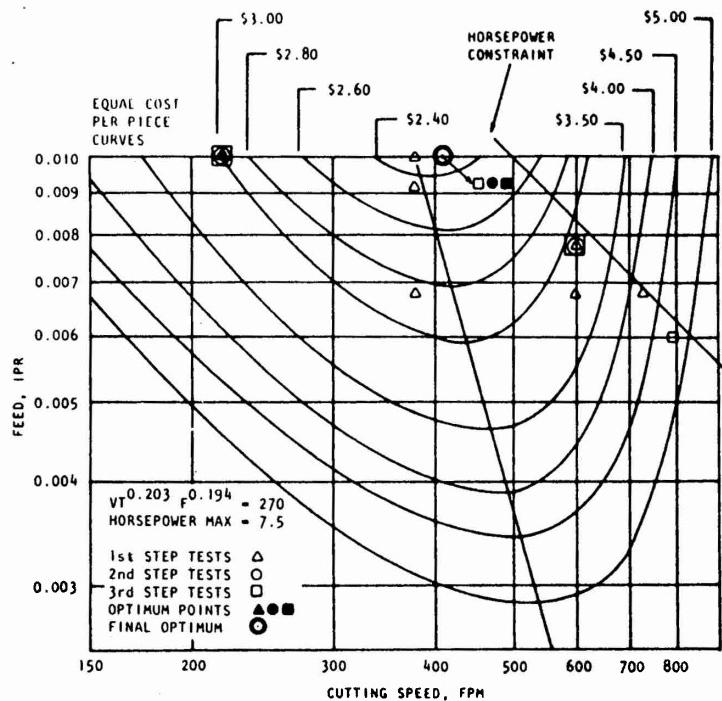
FIG. 5. Computer simulation Example 2, optimization analysis. Cost per piece index with 10% error, initial test points selected by computer, roughing condition.
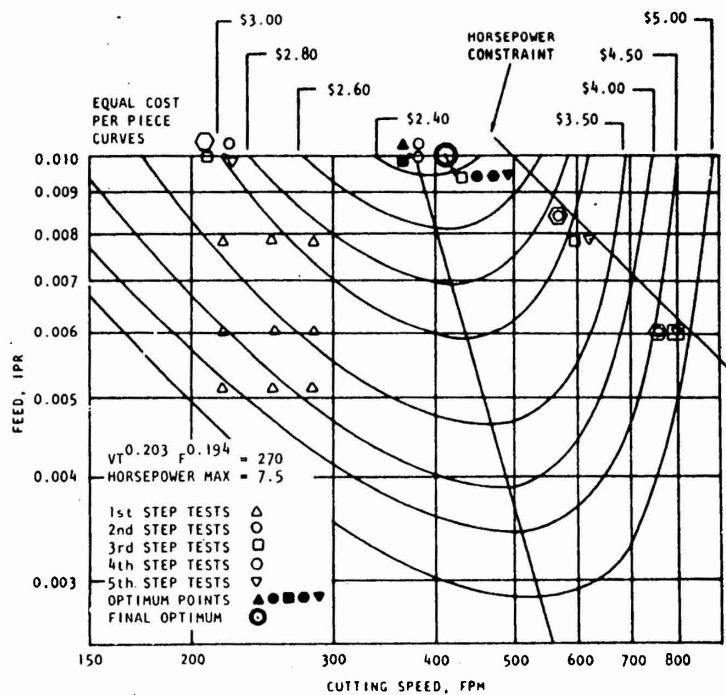
FIG. 6. Computer simulation Example 3, optimization analysis. Cost per piece index with 5% error, manual selection of initial test points, roughing condition.
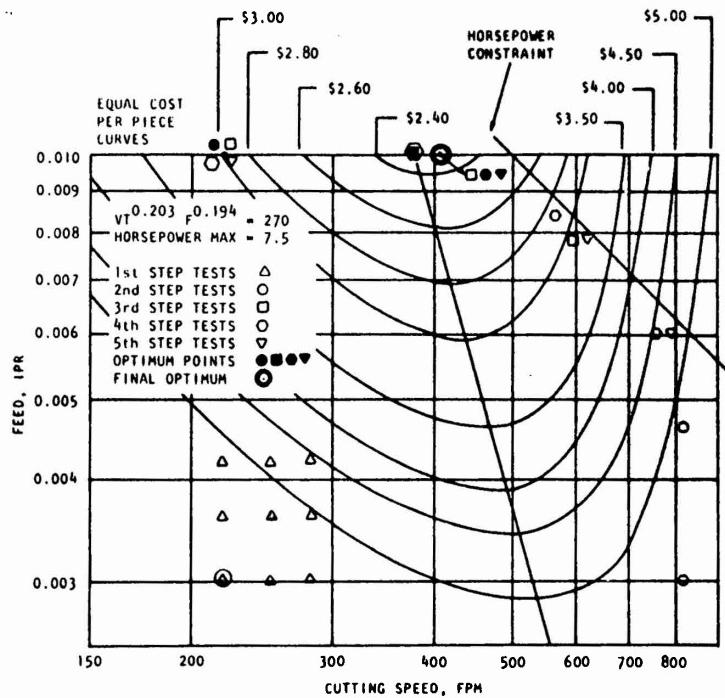
FIG. 7. Computer simulation Example 4, optimization analysis. Cost per piece index with 5% error, manual selection of initial test points far from expected optimum, feed rate extended to include finishing.
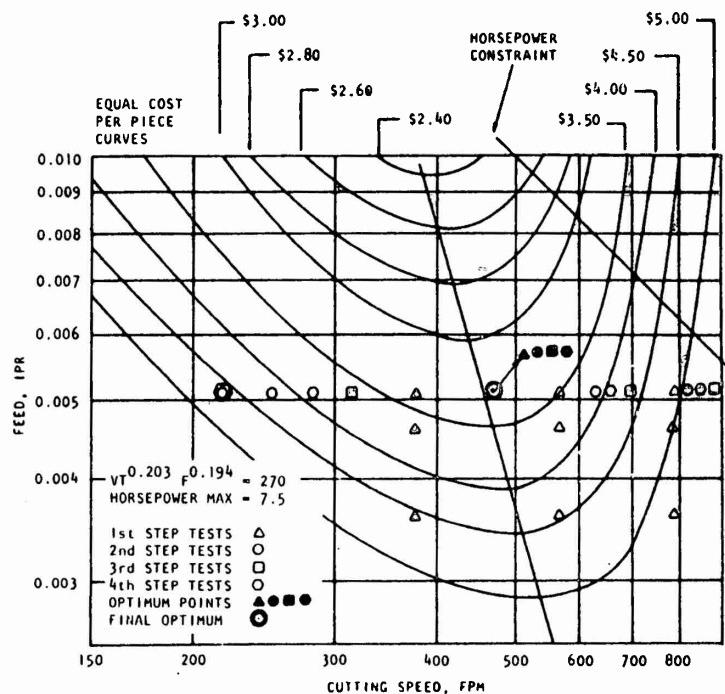
FIG. 8. Computer simulation Example 5, optimization analysis. Cost per piece index with 5% error, initial test points selected by computer, feed range restricted to finishing condition.

25

The results for this simulation are shown in Fig. 8. Note that because of the lower feed range the computer did not have to dodge the horsepower constraint as it did in Examples 1 and 2. It chose the nine initial points in the desirable geometric grid. This case took four steps of optimization analysis to reach the final recommendation of 300 rpm or 471.2 fpm and 0.0051 ipr.

2.5 PRODUCTION APPLICATION

When the PIM program is used to optimize an actual production case, all of the initial input is real shop data and each test point proposed by the computer is actually tested in the shop. The results of the machining tests, fed back to the program, are the basis for the program's stepwise estimates of optimums and its selection of additional points to be tested until the final optimum point is found.

Described in this section is a "production run" of the PIM program, with the Penn State Machinability Laboratory and its facilities serving as the shop.

NOTES ON PROCEDURE

Program Adjustments. As the PIM program is shown in Appendix A, it includes some cards that apply only to the simulation procedure for Section 2.4. For a production run, all cards with C* in the first two columns must be converted to active FORTRAN IV statements. That is done by reproducing those cards, leaving blanks in place of C*. Also, any card marked for removal from the deck must be removed. When these two adjustments are made, the card deck will be ready to run with real shop data.

Since cost data are supplied in the initial input and tool life and time data are generated by the shop tests, the MACSIM subroutine used in the simulations is not needed in production applications.

Selection of Initial Test Points. One of the basic functions of the PIM program is to make an efficient choice of initial test points. It performs this function in the production example.

The user may, however, choose the initial test points himself (Section 2.4, Example 3). In that case he must collect all of the information needed for determining the proper test values. He must know what constraints will affect the choice, because he cannot pick points outside these constraints. He must know what speed and feed settings are available. And he must be sure to choose at least six points, preferably as many as nine, spaced several discrete settings apart and arranged in a geometric pattern (Section 2.2, Step 1).

If the machine tool to be used is stepless, artificial steps will have to be calculated because PIM computation depends upon discrete values of the speed and feed parameters. The computer program will calculate these steps if the speed range and the desired step size are designated in the data deck. But the user, if he wishes, may calculate the steps and read them into the deck.

26

PRODUCTION TESTS

However the initial test points are chosen, the result will be six or more sets of cutting speed and feed settings, each of which must be tested on the machine tool. As the optimization analysis proceeds, the computer will recommend additional points to be tested in the shop. To avoid the effects of error, which show as oscillations of the optimum, it is important to keep the test conditions as consistent as possible.

Each test should be run for the longest period of time that is practical. The time period of a test should be at least several times greater than both the tooling time for each piece and the life of the cutting tool. The periods do not have to be equal, but no one test period should be conspicuously shorter than the others. If some unusual event, such as a long machine breakdown or an operator delay, interrupts the test period, the lost time should be subtracted from the total time for the test. Brief interruptions that are normal in production operations should not be deducted.

For each set of speed and feed settings tested, the operator will return three numbers to the program for further analysis:

$$T(I) = \text{time period of test, min}$$

$$P(I) = \text{number of pieces produced}$$

$$TP(I) = \text{number of tool changes}$$

When a piece is not completed or a cutting tool is still in use at the beginning or end of the period, the piece or the tool can be either counted or left out, according to the operator's judgment.

PRODUCTION EXAMPLE: TEST CONDITIONS

To put the PIM program to a realistic production test, the initial input information was taken from our "shop," the Machinability Laboratory. Computer runs were made at each step just as they would be in a plant machine shop, and the tool life tests were performed on a laboratory machinability lathe.

Operation and Tooling. The job, tooling, and material for the production example were as follows:

Operation	Straight turning
Machine tool	Engine lathe (IDENT = 1)
Workpiece	Custom-455 steel, 6 in. diam, 24 in. long
Cutting tool	C-2 carbide insert

Performance Index. As for the simulation examples, the performance index chosen was cost per piece, Q = 0. The cost data determined for this particular shop were

Labor and overhead rate	$0.10/min
Cutting tool cost	$0.20/edge

Machining Constraints. These constraints were determined by the characteristics of the lathe:

Maximum horsepower	7.5
Specific horsepower	$0.75/in.^3/min$
Machine efficiency	60%

(Note that the horsepower constraint is not essential to the program. It can be optional in plant applications, depending on the availability of the data.)

Available spindle speed range	20 to 1000 rpm in 50 steps
Available feed range	0.0011 to 0.0168 ipr in 24 steps

From past experience, it was possible to estimate for this material and this carbide cutting tool the practical operating range for rough machining:

Usable speed range	200 to 500 fpm (127 to 320 rpm)
Usable feed range	0.0051 to 0.0102 ipr
Depth of cut	0.10 in.

PRODUCTION EXAMPLE: INITIAL DATA INPUT

Having gathered these data, we are now ready to instruct the computer program to select the initial test points. The following data are read into the card deck (see Section 2.3 for code definitions):

IDENT = 1   M   = 24.

VLOW   = 127.0   F(I) = 0.0011 ⎫
                           to   ⎬ 24 steps
VHIGH = 320.0   0.0168 ⎭

FLOW   = 0.0051   KT   = 0 ⎫
FHIGH = 0.0102   IR   = 0 ⎪
                            ⎪ Zero on initial run.
VFLOW = 200.0   NPT = 0 ⎬ Computer supplies values
VFHIGH = 500.0   ⎪ for succeeding runs.
                   VMAX = 0 ⎪
SHP   = 0.75   FMAX = 0 ⎭

HPM   = 7.5   VL(I) ⎫
EFF   = 0.60   FD(I) ⎪ No cards needed for
DC   = 0.1   T(I) ⎬ initial run. NPT cards
                            ⎪ needed for succeeding
RLO   = 0.1   P(I) ⎪ runs.
TLC   = 0.2   TP(I) ⎭

Q   = 0.

D   = 6.

VI   = 20.0

VA   = 1000.0

VCH   = 20.0

28

PRODUCTION EXAMPLE: PIM ANALYSIS

The complete computer printout for the production example is given in Appendix A and the analysis is shown graphically in Fig. 9.

Because this example is intended to demonstrate how the program operates in a production application, the analytic process is described as a stepwise procedure with some general instructions and notes. The reader will find it useful to follow these steps in the computer printout, remembering that all numerical data apply to the example only.

Step 1: Initial Test Points and Tests. Based on the data input, the first output of the program is the nine initial test points, shown in the three columns headed "Test Conditions" in the following tabulation. These are actually combinations of three speeds and three feeds. For a time period of 240 min (4 hr) in this case, each combination was tested on the lathe, with these results:

Test Conditions			Production Test Results		
Cutting Speed	Speed	Feed,	Time,	No. of	No. of
rpm	fpm	ipr	min	Pieces	Tools
160	251	0.0068	240	11	1
160	251	0.0092	240	15	2
160	251	0.0102	240	16	2
220	346	0.0068	240	15	7
220	346	0.0092	240	20	9
220	346	0.0102	240	22	10
260	408	0.0068	240	18	16
260	408	0.0092	240	24	21
260	408	0.0102	240	27	23

The results are fed back to the computer program by adding the following cards:

NPT = 9 (Number of tests points run, IF)

$$
\left.\begin{array}{l}
VL(I) = \\
FD(I) = \\
T(I) = \\
P(I) = \\
TP(I) =
\end{array}\right\}
\begin{array}{l}
\text{Settings and results} \\
\text{for each test (9 cards)}
\end{array}
$$

Note that in real production the operator might have chosen, from his own experience, to eliminate the group of tests at the lowest cutting speed (160 rpm). He would then report only the remaining six tests, as NPT = 6. But at least six tests should be fed back for the regression analysis.

Note also that the test runs, though they vary in efficiency, are not "lost time" in production. During the 36 hr spent on the first tests for this example, 168 pieces were actually produced.
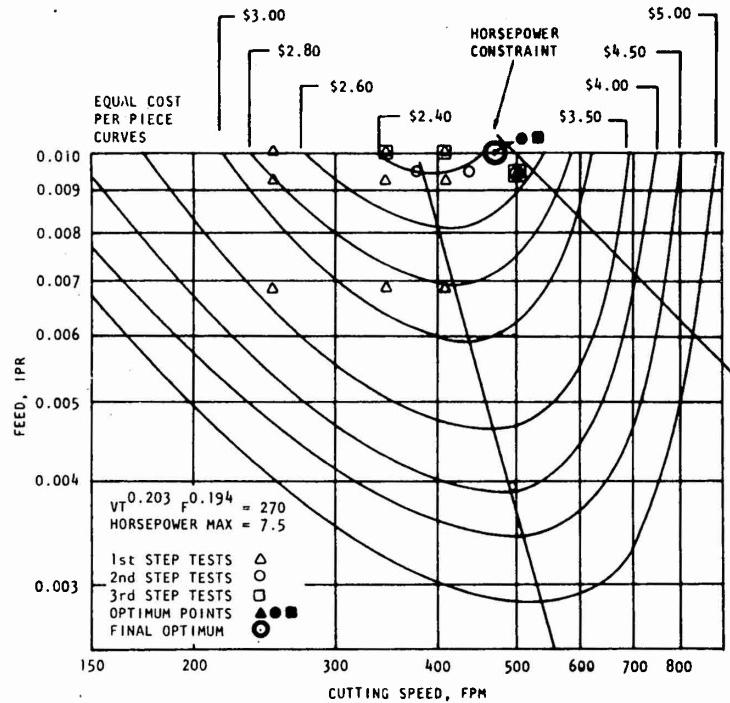
FIG. 9. Production example with shop test data.
Cost per piece index, initial test points select-
ed by computer, roughing condition.

Step 2: First Optimization. At the end of the first run, the com-
puter prints two codes for the next run. For this example they are

> IR = 0
> IF = 9

The IR code indicates whether the test results will be fitted to a sur-
face regression analysis (IR = 0) or to a curvilinear regression analysis
(IR = 1). The IF code indicates the number of test results (NPT) to be
reported. Along with the Step 1 test data, these codes must be returned
to the computer before the first optimization run of the program.

At this point we also activate the KT code to indicate the sequence
number of this analysis, as KT = 1. By raising the KT number with each
succeeding run (KT = 2 and so on) we get numbered printouts that can
prevent confusion, particularly when the number of optimization runs is
greater than two. Use of the KT numbers is optional, however; the pro-
gram can be run without them.

The second computer run produces the first optimum condition for the
chosen performance index, cost per piece:

> Cutting speed   320 rpm or 502.7 fpm
>
> Feed   0.0094 ipr

30

It then recommends, in this case, three conditions for further testing (shown in the following tabulation). *One of these conditions will always be the optimum condition previously computed.* The printout again ends with the codes for the next run.

The three new points were tested on the lathe, with due concern for the cautions indicated in the paragraphs headed PRODUCTION TESTS. These tests yielded the following feedback to the computer (three cards):

Test Conditions			Production Test Results		
Cutting Speed		Feed,	Time,	No. of	No. of
rpm	fpm	ipr	min	Pieces	Tools
240	377	0.0094	240	23	14
280	440	0.0094	240	26	31
320	503	0.0094	240	30	59

Step 3: Second Optimization. At the start of this run (KT = 2) we must again pick up the codes printed at the end of the Step 2 computer output:

IR = 1
IF = 3

This means that the regression analysis of the three new test points will be curvilinear.

On the next computer printout we get the second optimum point recommendation:

Cutting speed   300 rpm or 471.2 fpm

Feed   0.0102 ipr

This time the computer recommends four conditions for additional testing, one of which is the second optimum. The codes given for the next run are IR = 1, IF = 4.

The tests were run on the lathe, and the following results were reported to the program:

Test Conditions			Production Test Results		
Cutting Speed		Feed,	Time,	No. of	No. of
rpm	fpm	ipr	min	Pieces	Tools
220	346	0.0102	246	22	10
260	408	0.0102	246	27	23
300	471	0.0102	246	31	47
320	503	0.0094	246	30	59

Note that the time period of these tests was increased, but it was kept consistent.

Step 4: Third Optimization. Carding in the codes from the end of the previous printout and the new data from Step 3, we again run the

program.   The result is the third recommended optimum:

Cutting speed   300 rpm or 471 fpm

Feed   0.0102 ipr

Stopping Criterion.   The optimum conditions computed by the last two runs are identical. The stopping criterion (Section 2.2, Step 6) has therefore been satisfied, and no further testing is indicated. If the stopping criterion had not yet been reached, the computer would have recommended a new set of test points and the procedure would continue as before.

It is important to note that the program user may exercise certain options as the computations approach the final optimum. If the last two optimums are only one or two settings apart, he may choose to end the procedure before the stopping criterion is met. The true optimum is probably on the continuum between those settings, and any of the two or three settings may be adequate for the accuracy desired. If the estimated optimum point oscillates among several close values, it may be reasonable to decide that any of these values is usable and the testing can be stopped.

CONCLUSION

We chose for this example a case for which the optimum cutting conditions had been established. The final optimum computed by the PIM program is very close to the known speed and feed figures, certainly close enough to support confidence that the program can be applied successfully to actual production if the test conditions are properly controlled.

PART 3. PRODUCTION OPTIMIZATION METHOD

When tool life data, cost data, and time study data are available, the production optimization method (POM) provides a simple and practical procedure for optimizing machining parameters. POM requires fewer shop tests than the performance index method (PIM), and it yields both more accurate optimization and a larger quantity of useful information. It is less versatile than PIM in that it is designed to analyze only turning operations, but it offers the industrial user greater flexibility in the selection of optimum conditions for metal cutting. For these reasons, POM is recommended as the better method for practical applications where the necessary data are obtainable.

Like the PIM program, the POM computer program is constructed to serve the machinist on the job. The mathematical processes described in Section 3.2 are written into it and will operate automatically unless they are modified by engineering decision. Output from the program is in the form of printed tables giving a ranked range of cutting conditions for roughing and finishing, keyed to specific production objectives.

Section 3.3 describes the program structure and the language code for input data. A sample application of POM is presented in Section 3.4.

3.1 CONCEPT AND TECHNIQUES

The production optimization method analyzes optimum cutting conditions by gradual refinement of tool life data fed back to the program from production tests in the workshop. By an exhaustive enumeration procedure the program computes and ranks all combinations of cutting speed, feed, and depth of cut within the constraints imposed by a given workpiece, machine tool, cutting tool, and surface finish requirement. It also orders these values in terms of one or more of three objective functions: minimum cost per piece, maximum production rate, and maximum profit rate (4).

TOOL LIFE DATA

The POM program has the unique characteristic of generating the required tool life data for any given material and cutting tool. This capability leads to a high degree of accuracy and flexibility in production optimization. It is also a distinct advantage in analyzing new workpiece and cutting tool materials.

POM employs the conventional concept of Taylor's tool life equation, in the general form of

$$VT^{\alpha}F^{\beta}D^{\gamma} = C$$

The analysis begins with an educated initial estimate of the tool life parameters for the specific case, based upon published tool life data or practical experience, or both. The values computed with this estimate are tested in production, and the test results are fed back to the program as data for further computation to refine the parameter values. This

process is repeated until the tool life equations resulting from two successive iterations show no difference, or a percentage of difference that falls within a specified limit. The accepted tool life equation then functions in the POM program's optimization procedure for the case being analyzed.

The production example in Section 3.4 demonstrates that inaccuracies in the estimated tool life parameters are cleared as the analysis progresses. If the initial guess proves to be exact or if the true tool life parameters are known at the outset and read into the equation, the program will quickly verify the values and go to the optimization procedure.

The mathematics involved in computing the tool life equations is explained in Section 3.2. The intention at this point is only to indicate the efficiency and simplicity of POM treatment as compared with the usual means of obtaining tool life information. Because every production machining job is influenced by plant factors, published tool life data, though abundant, can provide no more than approximations for specific cases. For new workpiece materials or new cutting tool materials, the published data are not likely to be helpful except for rough estimating.

When a tool life equation must be evaluated, the usual resort is to elaborate, time-consuming, and expensive tool life tests at a machinability laboratory. Such tests are performed under closely controlled conditions, with tool life criteria based upon a predetermined limit of either flank wear or crater wear. There is no guarantee that the resulting tool life data can be readily applied to production operations with a particular machine tool in a particular workshop. Unless they are well tested and adjusted to accommodate local restrictions, tool life data obtained in a laboratory often prove to be inaccurate in the shop (6,7,8,9).

Tool life data computed by the POM program are fitted to the individual case. The shop tests are performed in the course of normal production, the computations are automatic and rapid, and the results are ready for immediate use.

MATHEMATICAL TECHNIQUES

The mathematical technique for evaluating the tool life equation is to compute the exponential values and the constant by solving simultaneous equations with a minimum of data. At least three sets of data for speed, feed, and tool life are needed to compute α, β, and C. (Since depth of cut is usually the least critical of the tool life parameters, for this analysis we assume that $\gamma = 0$). The initial equations are solved with estimated values, which are then refined by additions of test data from the shop.

Conventionally, the method of partial derivatives is applied to optimization analysis of machining conditions. That method can optimize only one parameter at a time, either cutting speed or feed, with the other parameters held constant. It does not present a combination of all parameters for the optimum condition. Also, the cutting speed for the maximum profit rate cannot be determined explicitly by this method except for particular values of the tool life exponents.

34

POM analysis employs instead an exhaustive enumeration procedure, subject to a series of cutting speeds, feeds, and depths of cut. By multiregression analysis, the program selects a range of optimum cutting conditions ranked according to one of three production objectives. As the general program is now written (Appendix B), the final optimization automatically prints rankings for minimum cost per piece, maximum production rate, and maximum profit rate, for both roughing and finishing conditions.

3.2   MATHEMATICAL PROCEDURES

This section explains the calculations essential to the POM program, which include the creation of artificial steps for a stepless lathe, the computation of cutting speed increments, evaluation of the tool life equation, and the relations for the three production objectives. Since the necessary procedures are written into the program, the production user need not concern himself with the mathematics. The information herein will be needed, however, if the general program is revised for a closer fit to the requirements of an individual plant.

ARTIFICIAL SPINDLE SPEED STEPS

Like the PIM program, the POM computer program is designed to optimize cutting conditions for either a stepped lathe (IDENT = 0) or a stepless lathe (IDENT = 1). For a stepped lathe, the speed steps are simply read in as known values. For a stepless lathe, the program requires artificial steps, which can be calculated in the following manner.

The minimum and maximum spindle speeds for a given lathe are always known. The user will specify the size of increment desired between spindle speed steps. With these data, the number of steps can be calculated as

$$X = [(RPM_a - RPM_l)/RPM_{ch}] + 1.0 \qquad (12)$$

where X = number of increments, a decimal number

RPM_a = highest spindle speed available, rpm

RPM_l = lowest spindle speed available, rpm

RPM_{ch} = size of increment

But X expressed as a decimal number has to be changed to an integer L, as L = [X]. So we add 1 to X to make the integer L, the largest number in X, large enough to span the range of spindle speeds.

The possibility still exists, however, that L will be less than X. In that case, the computer program must automatically increase L by 1. The program therefore computes RPM_i, the values of the L steps, as

$$RPM_i = RPM_l + (I - 1)RPM_{ch} \qquad I = 1,2,\ldots L \qquad (13)$$

The range of I must begin with 1 because the computer cannot index from zero.

CUTTING SPEED INCREMENTS

Having set the incremental values of the spindle speed, we can now calculate the cutting speed increments and the minimum and maximum cutting speeds possible for this operation. Because cutting speed is a function of the workpiece diameter, we must know the initial and final diameters for these calculations:

$$V_{min} = \pi D_s (RPM_1)/12.0 \qquad (14)$$

where V_{min} = cutting speed, fpm, at initial diameter with lowest spindle speed (rpm)

D_s = initial diameter of workpiece, in.

RPM_1 = lowest spindle speed available, rpm

and

$$V_{max} = \pi D_f (RPM_a)/12.0 \qquad (15)$$

where V_{max} = cutting speed, fpm, at final diameter with highest spindle speed (rpm)

D_f = final diameter of workpiece, in.

RPM_a = highest spindle speed available, rpm

The cutting speed increments V_k can then be calculated with equations in the form of

$$V_k = V_{min} \exp (\log V_{max}/V_{min})^{(K-1)/(L-1)} \qquad K = 1,2,...L$$

This is a step function used to form L steps between V_{min} and V_{max}. Since $\exp (\log X) = X$, we can write the equations as

$$V_k = V_{min}(V_{max}/V_{min})^{(K-1)/(L-1)} \qquad K = 1,2,...L \qquad (16)$$

It follows that if K = 1,

$$V_1 = V_{min}(V_{max}/V_{min})^0 = V_{min}$$

and if K = L,

$$V_L = V_{min}(V_{max}/V_{min})^1 = V_{max}$$

PARAMETERS OF TOOL LIFE EQUATION

Since depth of cut D^γ appears to be the least significant factor for this analysis, we can reduce the Taylor tool life equation to

$$VT^\alpha F^\beta = C \qquad (17)$$

where V = cutting speed, fpm
T = tool life, min.
F = feed, ipr
α, β = exponents
C = constant

With this as our basic relation, we begin the iterative process of computing and refining the tool life parameters α, β, and C.

Four situations occur in every search for an optimum cutting condition: 1) no shop data are available, 2) one set of shop data is available, 3) two sets of shop data are available, and 4) three or more sets of shop data are available. The program deals with these situations stepwise as they occur.

Step 1: No Shop Data. To obtain an initial set of cutting speeds and feeds, the program is run without any shop data (NDATA = 0). The tool life equation for this step is

$$VT^{\alpha_1} F^{\beta_1} = C_1$$

for which α_1, β_1, and C_1 are estimated values, an "educated guess," supplied by the user.

With these data, the program generates 20 ranked sets of speed and feed combinations within the capability of the machine tool. The computer output is a printed table optimized for the given production objective by exhaustive enumeration. One set is selected from the table and tested in the shop to determine a tool life for that combination. The user may choose the first set or may exercise his own judgment in the selection. The shop test data, which should always be an average of several tests, are returned to the program.

Step 2: One Set of Shop Data. With one set of "known" conditions, only one of the three unknown tool life parameters can be calculated. We choose to calculate first the constant C. Calling the values from the shop tests V_1, F_1, and T_1, we find a new value of C from the relation

$$C_2 = V_1 T_1^{\alpha_2} F_1^{\beta_2} \qquad (18)$$

in which $\alpha_2 = \alpha_1$ and $\beta_2 = \beta_1$.

The tool life equation for Step 2 becomes

$$VT^{\alpha_2} F^{\beta_2} = C_2$$

and from this equation the computer program generates another 20 ranked sets of speeds and feeds. For one of these sets, shop tests will return to the program the data for V_2, F_2, and T_2. For mathematical reasons related to the logarithms used later, care must be taken to select a test set that will yield values such that $V_2 \neq V_1$ and $F_2 \neq F_1$.

Step 3: Two Sets of Shop Data. There are now two distinct sets of data, V_1, F_1, T_1 and V_2, F_2, T_2, allowing two of the three tool life parameters to be calculated. We choose C and β. The new values C_3 and β_3 must satisfy both sets of data. That is,

$$V_1 T_1^{\alpha_3} F_1^{\beta_3} = C_3 \qquad V_2 T_2^{\alpha_3} F_2^{\beta_3} = C_3$$

in which $\alpha_3 = \alpha_2$. Since both of the left sides equal C_3, they must be equal:

$$V_1 T_1^{\alpha_3} F_1^{\beta_3} = V_2 T_2^{\alpha_3} F_2^{\beta_3}$$

or

$$(F_1/F_2)^{\beta_3} = (V_2/V_1)(T_2/T_1)^{\alpha_3}$$

We take logarithms of both sides of that equation:

$$\log (F_1/F_2)^{\beta_3} = \log [(V_2/V_1)(T_2/T_1)^{\alpha_3}]$$

By the rules of logarithms, the following changes can be made:

$$\beta_3 \log (F_1/F_2) = \log (V_2/V_1) + [\alpha_3 \log (T_2/T_1)]$$

$$= - \log (V_1/V_2) - [\alpha_3 \log (T_1/T_2)]$$

We then solve for β_3:

$$\beta_3 = [- \log V_1/V_2 - (\alpha_3 \log T_1/T_2)]/\log F_1/F_2 \qquad (19)$$

Having obtained a value for β_3, we can solve for C_3:

$$C_3 = V_2 T_2^{\alpha_3} F_2^{\beta_3} \qquad (20)$$

The new tool life equation

$$VT^{\alpha_3} F^{\beta_3} = C_3$$

is used to generate 20 more ranked combinations of speeds and feeds. Again, the one set selected to test in the shop for T_3 must be such that $V_3 \neq V_2 \neq V_1$ and $F_3 \neq F_2 \neq F_1$. The data from these tests are returned to the computer program as V_3, F_3, T_3.

38

Step 4: Three or More Sets of Shop Data. With at least three sets of data, we can calculate all three of the tool life parameters. The procedure described for Step 4 can be used only when there are three or more sets of shop data (n = 3,4,5...).

By multiple regression analysis, we will seek a linear equation that best describes the discrete data points available. Taking the logarithm of both sides of Eq. 17 and simplifying it, we get a linear equation suitable for regression analysis:

$$\log V + (\alpha \log T) + (\beta \log F) = \log C$$

in which log V is the dependent variable, log T and log F are the independent variables, and α, β, and C are constants. Rewritten in standard form, this equation becomes

$$\log V = \log C - (\alpha \log T) - (\beta \log F)$$

If log V is taken to be a value from the desired equation and the right side an estimated value from the data, the object of this analysis is to minimize the difference between the two:

$$\log V - [\log C - (\alpha \log T) - (\beta \log F)]$$

There are n such equations, of course, and the difference should be minimized for all n points. Since the differences D may be plus or minus, we will use the sum of their squares:

$$D = \sum_{i=1}^{n} [\log V_i - \log C + (\alpha \log T_i) + (\beta \log F_i)]^2$$

If the partial derivatives of D with respect to the three constants are taken and set equal to zero, the resulting values of the constants will ensure the minimum possible value of D. That is,

$$\partial D/(\partial \log C) = -2 \sum_{i=1}^{n} [\log V_i - \log C + (\alpha \log T_i) + (\beta \log F_i)] = 0$$

$$\partial D/\partial \alpha = 2 \sum_{i=1}^{n} [\log V_i - \log C + (\alpha \log T_i) + (\beta \log F_i)] \log T_i = 0$$

$$\partial D/\partial \beta = 2 \sum_{i=1}^{n} [\log V_i - \log C + (\alpha \log T_i) + (\beta \log F_i)] \log F_i = 0$$

Rewriting the first of these equations by dividing out the -2 and distributing the summation signs gives

$$\sum_{i=1}^{n} \log V_i - \sum_{i=1}^{n} \log C + (\alpha \sum_{i=1}^{n} \log T_i) + (\beta \sum_{i=1}^{n} \log F_i) = 0$$

which can be rearranged with $\sum_{i=1}^{n} \log C = n \log C$, so that

$$n \log C - (\alpha \sum_{i=1}^{n} \log T_i) - (\beta \sum_{i=1}^{n} \log F_i) = \sum_{i=1}^{n} \log V_i \qquad (21)$$

Likewise, the second and third equations can be written as

$$\log C \sum_{i=1}^{n} \log T_i - [\alpha \sum_{i=1}^{n} (\log T_i)^2] - (\beta \sum_{i=1}^{n} \log F_i \log T_i)$$

$$= \sum_{i=1}^{n} V_i \log T_i \qquad (22)$$

$$\log C \sum_{i=1}^{n} \log F_i - (\alpha \sum_{i=1}^{n} \log T_i \log F_i) - [\beta \sum_{i=1}^{n} (\log F_i)^2]$$

$$= \sum_{i=1}^{n} \log V_i \log F_i \qquad (23)$$

We now have expressions for each of the three unknowns, Eqs. 21, 22, and 23. These equations can be solved in any one of a great number of ways. The POM program employs determinants to find the values of log C, α, and β. The tool life constant C is then found by

$$C = \exp (\log C)$$

where log C is what is known.

The third set of shop data provides the first set of calculated tool life parameters: α_4, β_4, C_4. When the fourth set of shop data is analyzed, the new parameters α_5, β_5, C_5 are compared to the first set to determine the percentage of change in the values:

$$\alpha = |(\alpha_5 - \alpha_4)/\alpha_4| \times 100$$
$$\beta = |(\beta_5 - \beta_4)/\beta_4| \times 100$$
$$C = |(C_5 - C_4)/C_4| \times 100$$

This procedure is continued until the value for each of the three parameters is equal to or less than the prescribed stopping criterion for that parameter. The resulting tool life equation is used to generate the final set of solutions.

PRODUCTION OBJECTIVES

Though job work with POM will usually be done in terms of a single production objective, the program as it is now written (Appendix B)

computes all three production objectives in the final optimization. The following are the relations for calculating the objectives.

1) <u>Minimum Cost per Piece</u>. This computation involves, of course, all of the cost, time, and machining data supplied to the program. The relation is

$$C_u = C_o[N(2T_1 + XL/F_r) + T_L + N_{gc}T_{gc} + T_c] + (C_oT_{ct} + C_e)P$$

where C_u = total machining cost, \$/piece

C_o = labor and overhead rate, \$/min

N = number of cuts

T_1 = time for motion at start and end of cut, min

XL = length of workpiece, in.

F_r = return speed of carriage, in./min

T_L = work handling time, min

N_{gc} = number of gear changes

T_{gc} = time per gear change, min

T_c = cutting time, min

T_{ct} = tool changing time, min

C_e = tool cost, \$/edge

P = number of edges per piece

2) <u>Maximum Production Rate</u>. The total time per piece T_{prod}, in minutes, is the sum of all the time elements:

$$T_{prod} = N(2T_1 + XL/F_r) + T_L + N_{gc}T_{gc} + T_c + T_{ct}P$$

The production rate, expressed as $60/T_{prod}$, is the reciprocal of the time per piece multiplied by 60 min, so that it is found as pieces per hour.

3) <u>Maximum Profit Rate</u>. This simple calculation relates all of the unit cost and time factors to the price per piece:

$$Profit = (Price - C_{mt} - C_u)/T_{prod}$$

in which Price is the selling price per piece, C_{mt} is the material cost, and the other factors are as previously defined.

3.3 COMPUTER PROGRAM FOR <u>POM</u>

For the production optimization method, as for the performance index method, the computer program uses the language of FORTRAN IV and was developed with the IBM 360, Model 67 computer in the Computation Center at Penn State. The POM program is adaptable, however, to other computer systems.

41

Since the mathematical operations explained in Section 3.2 are written into the program and the code terms for data input are directly related to the vocabulary of production, POM analysis can be performed by shop personnel who are not familiar with the mathematical techniques or with computer programming.

The structure of the program is described in detail in Appendix B, which provides the flow diagram, the format guide for data input cards, and a complete listing of the computer program. How POM operates is briefly surveyed in this section and demonstrated by the production example in Section 3.4.

For ready reference in relation to the production example, the input data code is listed and defined at the end of this section. The code terms are given in the order of their occurrence in the format guide in Appendix B.

GENERAL PROCEDURE

The POM program uses cost data, time study data, machine parameters, and tool life data obtained in shop tests to arrive at an optimum set of cutting conditions for any one or all of three production objectives. The initial data, including an estimate of tool life parameters for the job, are gathered and prepared for the computer according to the format guide (Appendix B). From this input the computer generates with subroutine RANK, explained later in this section, a ranked set of speeds and feeds.

The user selects one of these combinations of feed and speed to test for tool life in his shop. Note that these tests are run as normal production operations, the only requirement being that no factor can be changed during several repetitions of the test for any one combination. The object is to obtain good average figures for the tool life parameters, which will be fed back to the program for refinement of the initial tool life estimate. The results of this shop test are punched into a data card (type 10), and this card is stored behind the original data, to be used in the second analytic run of the program.

That procedure is repeated, adding one data card per investigation, until the change in each of the tool life equation parameters is smaller than some acceptable value. The final optimization analysis is then generated with subroutine OPTM, described later herein.

SPINDLE SPEED STEPS

When all of the job constants (cards 1 through 3) have been read into the program, the computer tests IDENT to determine the type of spindle speed control this machine tool has. If IDENT = 0, the lathe is stepped and the computer looks for two or more cards, one giving the number of speed steps and one or more giving the values of the spindle speed at each step.

If IDENT = 1, the lathe is stepless and the computer looks for a single data card containing the lowest and highest spindle speeds available and the size of the increment from one speed to the next (card 4), as

42

specified by the user. With these data, the computer will calculate the number of artificial speed steps and the spindle speed at each step (Eq. 13).

CUTTING SPEED AND FEED RATE

From the data for the possible number of feed settings and the feed rate at each setting, the program is then able to compute V_{min} and V_{max} and the set of stepped cutting speeds for this operation (Eqs. 14, 15, 16).

Next, the constraints on the feed rate are established. If the user elects to limit the range of feeds suitable for this cutting operation, he simply inputs with the initial data the lowest and highest feeds he wishes to investigate, as FLOW and FHIGH. But if the range is to include all available feeds, FLOW and FHIGH need not be specified in the input data. In that case, the program takes the first feed read in to be FLOW. The feeds must be ordered from low to high on the input data cards.

Constraints on the cutting speed are established in exactly the same way as the constraints on the feed rate.

TOOL LIFE PARAMETERS

As soon as the initial tool life data (card 9) are read into the computer program as the next step, all data supplied or calculated up to this point are checked for validity. If a discrepancy is detected, an error message is printed and the program is stopped until the defect has been remedied.

When the program proceeds, NDATA will be zero on the first run and will be incremented by 1 on each succeeding run. As the shop test data are received (card 10), the program systematically improves the tool life equation by determining the number of data points available and applying the appropriate mathematical steps to obtain one more calculated parameter or to improve all of the parameters ALPHA, BETA, and C (see Section 3.2, Steps 1 through 4).

When the value computed for each of the tool life parameters is equal to or less than the values specified by PERALF, PERBET, and PERC, the program accepts the tool life equation. Subroutine OPTM is then called, and the refined tool life equation is used in generating the final output of the POM computer analysis.

SUBROUTINE RANK

For any given set of tool life parameters, this subroutine provides a ranked set of optimum feeds and speeds for the production objective specified as ICRIT in the main program (card 9). ICRIT offers three choices, as shown in the definition in the code list at the end of this section. By exhaustive enumeration, RANK searches all possible values of feed, cutting speed, and depth of cut to find those that best satisfy the optimization criterion. It takes into account all constraints read into the program for the job, using only allowable values. The mathematical relations for these calculations are given at the end of Section 3.2.

The output of RANK each time it is called is two printed tables, one showing the range of feeds for roughing, the other the range for finishing. Each of these tables lists only the 20 best optimizing values of cutting speed, feed, and depth of cut. From one or the other of the tables, depending on his purpose, the user selects the values of feed and cutting speed for the next step of the analysis. He would logically choose the set of F and V values ranked first, but ONLY IF NEITHER HAS BEEN TESTED PREVIOUSLY. If one or both of those values occurred in an earlier data set, he will search down the list for a set of distinctly new values and choose that set.

SUBROUTINE OPTM

This subroutine operates in the same manner as RANK, but it is called only when the acceptable values of tool life parameters ALPHA, BETA, and C have been found. As the general POM program is written (Appendix B), the table printouts from OPTM give the optimum values of feed, cutting speed, and depth of cut for all three of the production objectives possible under ICRIT. Again, there are two sets of ranked tables, one for roughing conditions and one for finishing conditions.

Though a minor amount of computer time can be saved by running OPTM with only a single optimization criterion, the long-range advantage of obtaining all of the information quickly and simply in the first analysis of a machining job should not be overlooked. The OPTM printouts take little filing space, and the necessary information will be already at hand if the same operation should recur with a change of production objective. This versatility is particularly desirable when new materials and new cutting tools are being analyzed.

INPUT DATA CODE

The following tabulation lists and defines the code used for input data in the POM program, with the corresponding mathematical notation.

Input Code	Notation	Definition
NPR		Number or code to identify project.
IDENT		Type of lathe: = 1 if speed is stepless. Computer calculates artificial steps. = 0 if speed is stepped. Steps are read in on data cards.
DCMIN	DC_{min}	Minimum depth of cut, in.
DCMAX	DC_{max}	Maximum depth of cut, in.
VLOW	V_{low}	Lowest cutting speed, fpm.
VHIGH	V_{high}	Highest cutting speed, fpm.
FLOW	F_{low}	Lowest feed, ipr.

Input Code	Notation	Definition
FHIGH	F_{high}	Highest feed, ipr.
XL	XL	Length of workpiece, in.
DIAS	D_s	Initial diameter of workpiece, in.
DIAF	D_f	Final diameter of workpiece, in.
FR	F_r	Return speed of carriage, in./min.
T1	T_1	Time for forward or backward motion at start or end of cut, min.
TL	T_L	Time for loading or unloading workpiece, min.
TCT	T_{ct}	Tool changing time, min.
TGC	T_{gc}	Gear changing time, min.
PRICE	SP	Selling price, $/piece.
COSTMT	C_{mt}	Material cost, $/piece.
FEEDRF	F_{RF}	Lowest feed rate for roughing, ipr.
CO	C_o	Labor and overhead rate, $/min.
CE	C_e	Tool cost, $/edge.
SHP	S_{hp}	Specific horsepower, hp/in.^{3}/min.
HPM	HP_m	Maximum horsepower.
EFF	eff	Efficiency of lathe, decimal form of percent.

[Supply the following three numbers only if IDENT = 1.]

Input Code	Notation	Definition
RPMI	RPM_1	Lowest possible speed of lathe, rpm.
RPMA	RPM_a	Highest possible speed of lathe, rpm.
RPMCH	RPM_{ch}	Desired size of artificial steps, rpm.

[The following two variables apply only if IDENT = 0.]

Input Code	Notation	Definition
L	L	Number of speed steps.
RPM(I)	RPM_i	Values of L steps, rpm. Must be ordered from low to high.
M	M	Number of feed steps.
FEED(I)	F_i	Values of M steps, ipr. Must be ordered from low to high.

(CONTINUED)

45

Input Code	Notation	Definition
NDATA		Code for number of data points available. NDATA = 0 for first run of program.
ICRIT		Code for desired optimization criterion: 1 = minimum unit cost. 2 = maximum production rate. 3 = maximum profit rate.
NVTYPE		Code telling how values of cutting speed VEL will be read in: 1 = cutting speed in fpm. 2 = cutting speed in rpm.
ALPHA	α	Estimated or previously calculated value of tool life exponent.
BETA	β	Estimated or previously calculated value of feed exponent.
C	C	Estimated or previously calculated value of tool life constant.
PERALF	α_{ch}	Stopping criterion for α. Accept $\alpha \leq \alpha_{ch}$.
PERBET	β_{ch}	Stopping criterion for β. Accept $\beta \leq \beta_{ch}$.
PERC	C_{ch}	Stopping criterion for C. Accept $C \leq C_{ch}$.
VEL(I)		Cutting speed or spindle speed tested in shop. See NVTYPE.
F(I)		Feed rate tested in shop.
T(I)		Tool life determined in shop for given speed and feed.

3.4 PRODUCTION EXAMPLE

This section demonstrates the procedure and results for an actual application of the production optimization method of analyzing optimum cutting conditions. Some sample tool life data obtained in the Machinability Laboratory at Penn State were used for computer simulation of production conditions.

PRODUCTION INFORMATION

The following initial information was gathered for the production example, based on the given job, the available tooling, and production experience in this "machine shop."

Operation and Tooling. These were the job and tool specifications:

    Operation   Straight turning
    Machine tool   Engine lathe (IDENT = 1)
    Workpiece   INCONEL-718 (Bhn-277), 24 in. long
    Cutting tool   C-2 carbide insert
    Diameter at start   8 in.
    Diameter at finish   7.5 in.

The tool geometry of the cemented carbide insert selected for this operation was 5-5-7-7-6-0-1-/16, U.S. standard.

Time Factors. From experience, the machining time factors were established as:

    Return speed of carriage   5 in./min
    Machine handling time   0.5 min
    Work handling time   5 min
    Tool changing time   1 min
    Gear changing time   0.5 min
       for different rpm

Cost Factors. The labor and overhead and tool cost figures were based on averages for this university machine shop. No attempt was made to approximate industrial figures, which may vary considerably. The material cost and selling price are assumptions based on current market values and practices.

    Labor and overhead rate $0.20/min
    Cutting tool cost $0.50/edge
    Material cost $300/piece
    Selling price $600/piece

Machining Constraints. The constraints were determined, of course, by the lathe characteristics and the nature of the workpiece and cutting tool materials.

    Maximum horsepower   7.5
    Specific horsepower   1.5/in.^{3}/min
    Machine efficiency   60%

    Available spindle speed range   20 to 1000 rpm in 50 steps
    Available feed range   0.0011 to 0.0168 in 24 steps

    Usable cutting speed   80.0 to 200.0 fpm
    Usable feed: finishing   0.003 to 0.008 ipr
                 roughing   0.008 to 0.0168 ipr

    Depth of cut   0.05 to 0.25 in.

Tool Life Parameters. For this demonstration we assumed that no data existed for the parameters of the Taylor tool life equation as it is used in POM analysis, $VT^\alpha F^\beta = C$ (Eq. 17). The following figures were

decided upon as the best initial guess, to be refined by the shop tests:

Tool life exponent $\alpha_1 = 0.40$
Feed exponent $\beta_1 = 0.40$
Tool life constant $C_1 = 15.0$

It was also decided that a reasonable error allowance (PERALPH, PERBET, PERC) for final acceptance of the tool life equation parameters would be 5%.

Other Specifications. Initial decisions had to be made on two other points. It was necessary to specify the production objective, and the choice for this case was minimum cost per piece (ICRIT = 1).

Since cutting speed (fpm) can also be given as spindle speed (rpm), we had to select the unit to be used for cutting speed values from the shop tests. We chose FPM (NVTYPE = 1).

INITIAL DATA INPUT

We now had all of the figures needed for initial data input to the POM program. The following data were read into the card deck in the order prescribed by the format guide (Appendix B). The code terms are defined at the end of Section 3.3.

IDENT	= 1		SHP	= 1.5
DCMIN	= 0.05		HPM	= 7.5
DCMAX	= 0.25		EFF	= 0.60
VLOW	= 80.0		RPMI	= 20.0
VHIGH	= 200.0		RPMA	= 1000.0
FLOW	= 0.003		RPMCH	= 20.0
FHIGH	= 0.0168		M	= 24
XL	= 24.0		FEED(I)	= 0.0011 to 0.0168 } 24 steps
DIAS	= 8.0			
DIAF	= 7.5		NDATA	= 0 (Initial run)
FR	= 5.0		ICRIT	= 1
T1	= 0.5		NVTYPE	= 1
TL	= 5.0		ALPHA	= 0.4 }
TCT	= 1.0		BETA	= 0.4 } Initial run
TGC	= 0.5		C	= 15.0 }
PRICE	= 600.0			
COSTMT	= 300.0		PERALPH	= 0.05
FEEDRF	= 0.008		PERBET	= 0.05
CO	= 0.2		PERC	= 0.05
CE	= 0.5			

COMPUTER PRINTOUTS

Completing the optimization of this example required five steps of computer analysis and four periods of shop tests. The printouts from each of the first four runs of the computer program consisted of three sections. The first section gave all of the job constants, which were the same for each run, plus the tool life conditions pertaining to the particular run. For the specified production objective, minimum cost per piece, the second section ranked the 20 best combinations of parameters for finishing conditions. The third section ranked the 20 best combinations for roughing conditions.

The printout for the fifth and final computer run consisted of seven sections. The first section repeated the specified conditions and all of the accumulated tool life data, and gave the final tool life parameters and the basis for accepting them. Then followed two sections for each of the three possible production objectives, one ranking the 20 best final optimizations for finishing conditions, the other the 20 best final optimizations for roughing conditions.

For reasons of economy in producing this publication, we show in Appendix B only three samples from the printouts, selected to indicate their general content and arrangement. Reproduced there are the first section and the finishing table from Step 2, and the final optimization for minimum cost per piece, finishing, from Step 5. A glance at the samples at this point will give the reader a concrete picture of the form and scope of the computer output, which will be useful as we proceed.

ANALYTIC PROCEDURE

Steps 1 through 4 in the following discussion correspond to the numbered steps under PARAMETERS OF TOOL LIFE EQUATION in Section 3.2. Step 5 is the final optimization.

Step 1: No Shop Data. The estimated tool life equation for the first run of the computer program is

$$VT^{0.4}F^{0.4} = 15.0$$

With that equation, the first output of the program is the two tables for finishing and roughing. In this discussion we will follow the optimization for only the finishing condition. Since we have no reason to do otherwise, we select from the table for that condition the first ranking combination to test on our shop lathe with the given material and cutting tool:

$$V_1 = 85 \text{ fpm}$$

$$F_1 = 0.0078 \text{ ipr}$$

$$D_1 = 0.250 \text{ in. (one pass)}$$

As the average of several tests employing that combination without change, the machinist returns to the program a tool life value of

$$T_1 = 26.3 \text{ min}$$

This value can be determined only by the operator in the production shop, since he determines the time for tool changes.

Step 2: One Set of Shop Data. We now have NDATA = 1 (card 9), the first set of shop data V_1, F_1, T_1 to enter into the program (card 10). Since one set of shop data permits us to compute only one of the tool life parameters, we decide to hold ALPHA and BETA at 0.4 and find a new value C_2 for the tool life equation constant, using T_1 = 26.3 min. The result of this computation is

$$C_2 = 45.1$$

and the tool life equation becomes

$$VT^{0.4}F^{0.4} = 45.1$$

With this equation, subroutine RANK produces another pair of tables. From the table for finishing (shown in Appendix B) we again select a combination to be tested in the shop. But we must remember that the values for both V and F must be different from the values previously tested (Section 3.2, Step 2). The first three feed values in the table are all F = 0.0078. The fourth combination shows no duplication of either V or F, and therefore seems suitable for testing. So the combination sent to the shop is

$$V_2 = 136 \text{ fpm}$$

$$F_2 = 0.0068 \text{ ipr}$$

$$D_2 = 0.25 \text{ in. (one pass)}$$

The average figure returned to the program from this group of tests on the lathe is

$$T_2 = 7.0 \text{ min}$$

Step 3: Two Sets of Shop Data. When NDATA = 2, we can compute two of the tool life equation parameters. Still holding ALPHA at 0.4, we obtain these values for BETA and C:

$$\beta_3 = -0.44$$
$$C_3 = 2597.6$$

and the tool life equation becomes

$$VT^{0.4}F^{-0.44} = 2597.6$$

Compared to the value computed for C_2, the new value for C_3 looks rather unreasonable. But let us see what the program will do with it.

Scanning the finishing table computed with this equation, we find that the first combination showing unused values of both V and F is

$$V_3 = 147.0 \text{ fpm}$$

$$F_3 = 0.0060 \text{ ipr} \qquad D_3 = 0.25 \text{ in. (one pass)}$$

With that set of cutting conditions, the machinist reports an average tool life value of

$$T_3 = 6.7 \text{ min}$$

Step 4: Three Sets of Shop Data. Now that NDATA = 3, we can compute all three of the tool life equation parameters, ALPHA, BETA, and C, and the computer gives us

$$\alpha_4 = 0.30$$
$$\beta_4 = 0.53$$
$$C_4 = 17.3$$

The first thing we notice about these figures is that the normal operation of the program has automatically brought the value of the constant C back into the ballpark.

With the new tool life equation

$$VT^{0.30}F^{0.53} = 17.3$$

we get two more optimizing tables. But this time our choice from the finishing table does not necessarily have to be governed by $V \neq V_{1,2,3}$ and $F \neq F_{1,2,3}$. Because the second ranked set of conditions offers generally better production values and only a slightly higher cost per piece than the set ranked first in terms of the production objective, we choose the second set for the next group of shop tests:

$$V_4 = 126.0 \text{ fpm}$$
$$F_4 = 0.0078 \text{ ipr}$$
$$D_4 = 0.25 \text{ in. (one pass)}$$

This combination yields

$$T_4 = 7.1 \text{ min}$$

Step 5: Acceptance and Optimization. Calculating that set of values, the computer finds the tool life equation parameters to be

$$\alpha_5 = 0.30$$
$$\beta_5 = 0.53$$
$$C_5 = 17.3$$

which means that the tool life equation agrees exactly with the equation obtained in Step 4:

$$VT^{0.3}F^{0.53} = 17.3$$

The first section of the printout tells us that the 5% change limit specified for PERALF, PERBET, and PERC has been satisfied and the program is ready to call OPTM, the final optimizing procedure for all three of the possible production objectives.

51

RESULTS AND CONCLUSION

Taking the first-rank figures from each of the six tables produced by OPTM, the results of POM analysis of optimum cutting conditions for this example can be briefly summarized as follows:

Param-	Min. Unit Cost		Max. Prod. Rate		Max. Profit Rate	
eter	Finishing	Roughing	Finishing	Roughing	Finishing	Roughing
V, fpm	116	85	126	107	126	85
F, ipr	0.0078	0.0112	0.0078	0.0092	0.0078	0.0112
D, in.	0.25	0.25	0.25	0.25	0.25	0.25
T, min	9.25	13.9	7.12	8.98	7.12	13.9

That is a great deal of information to obtain from a little over one minute of computer time. Note also that the 20 choices provided by the complete tables for roughing and finishing offer the industrial user maximum flexibility.

In this case, as in the case of PIM, we tested an example for which the optimum cutting conditions had been determined by conventional means. Provided that the tool life data returned from the shop at each step are the average from several tests and the specifications for choosing the combination to be tested are observed, the results of POM analysis are quite acceptable for industrial application.

BIBLIOGRAPHY

1. Bhattacharyya, A., Faria-Gonzalez, R., and Ham, I., "Regression Analysis for Predicting Surface Finish and Its Application in the Determination of Optimum Machining Conditions," *Journal of Engineering for Industry, Trans. ASME,* Vol. 92, Aug. 1970, pp. 711-716.

2. Brown, R. H., "On the Selection of Economical Machining Rates," *International Journal of Production Research,* Vol. 1, Mar. 1962, pp. 1-22.

3. Crookall, J. R., "The Performance-Envelope Concept in the Economics of Machining," *International Journal of Machine Tool Design and Research,* Vol. 9, 1969, pp. 261-278.

4. Duncan, A. J., *Quality Control and Industrial Statistics,* 3rd ed., R. D. Irwin, Homewood, Ill., 1965.

5. Field, M., Zlatin, N., Williams, R., and Kronenberg, M., "Computerized Determination and Analysis of Cost and Production Rates for Machining Operations," Part I, ASME Paper 67-WA/Prod-18, 1967; Part II, ASME Paper 68-WA/Prod-17, 1968.

6. Ham, I., "Tooling Costs in Machining Economics," ASTME Tech. Paper MR69-282, 1969.

7. Ham, I., "Economics of Machining: Analyzing Optimum Machining Conditions by Computers," ASTME Tech. Paper SP64-60, 1964.

8. Ham, I., and Bhattacharyya, A., "Generalized Analysis for Optimization of Machining Conditions," SME Tech. Paper MR70-542, 1970.

9. Ham, I., and Faria-Gonzalez, R., "Production Optimization Method (POM) Using a Digital Computer," *Proceedings, International Machine Tool Design and Research Conference,* University of Birmingham, England, Sept. 1970, Pergamon Press, New York, 1971.

10. Hitomi, K., "Economical and Optimum Seeking Machining," *Metal Processing and Machine Tools,* Instituto per le Ricerche di Tecnologia Meccanica, Vico Canavese, Nov. 1970.

11. Peklenik, J., "Statistical Concept of Static and Dynamic Testing of Manufacturing Systems," SME Paper MM71-221, 1971.

12. Porter, B., and Summers, R. D. M. J., "The Performance of Self-Optimalizing Strategies in the Adaptive Control of the Metal-Cutting Process," *International Journal of Machine Tool Design and Research,* Vol. 8, 1968, pp. 217-237.

13. U.S. Air Force, *Machinability Reports,* Vol. I, 1950; Vol. II, 1951; Vol. III, 1954; Vol. IV, 1960.

14. U.S. Army Material Command, *Machining Data Handbook,* 1966.
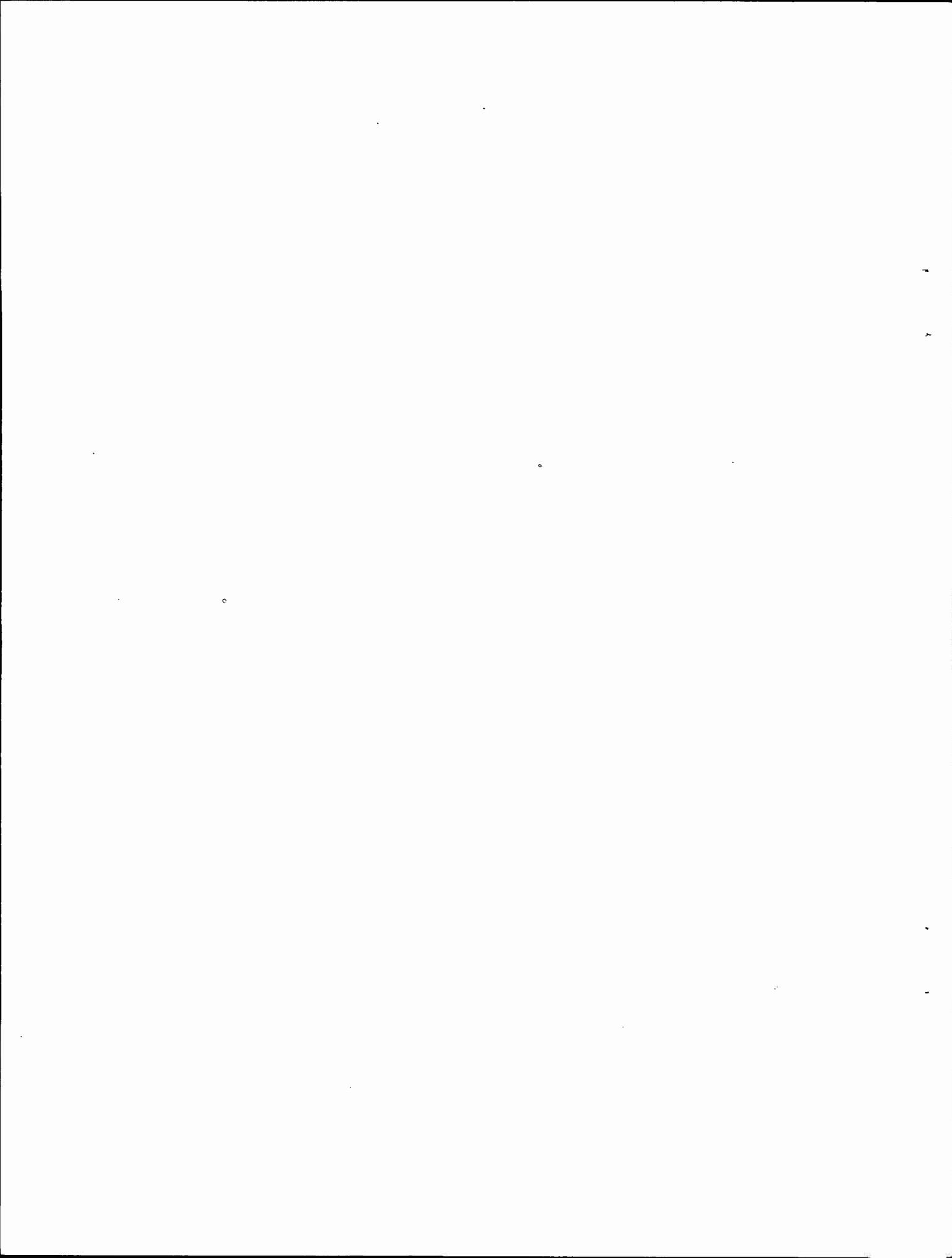
APPENDIX A. PERFORMANCE INDEX METHOD

55

FLOW DIAGRAM, PIM COMPUTER PROGRAM
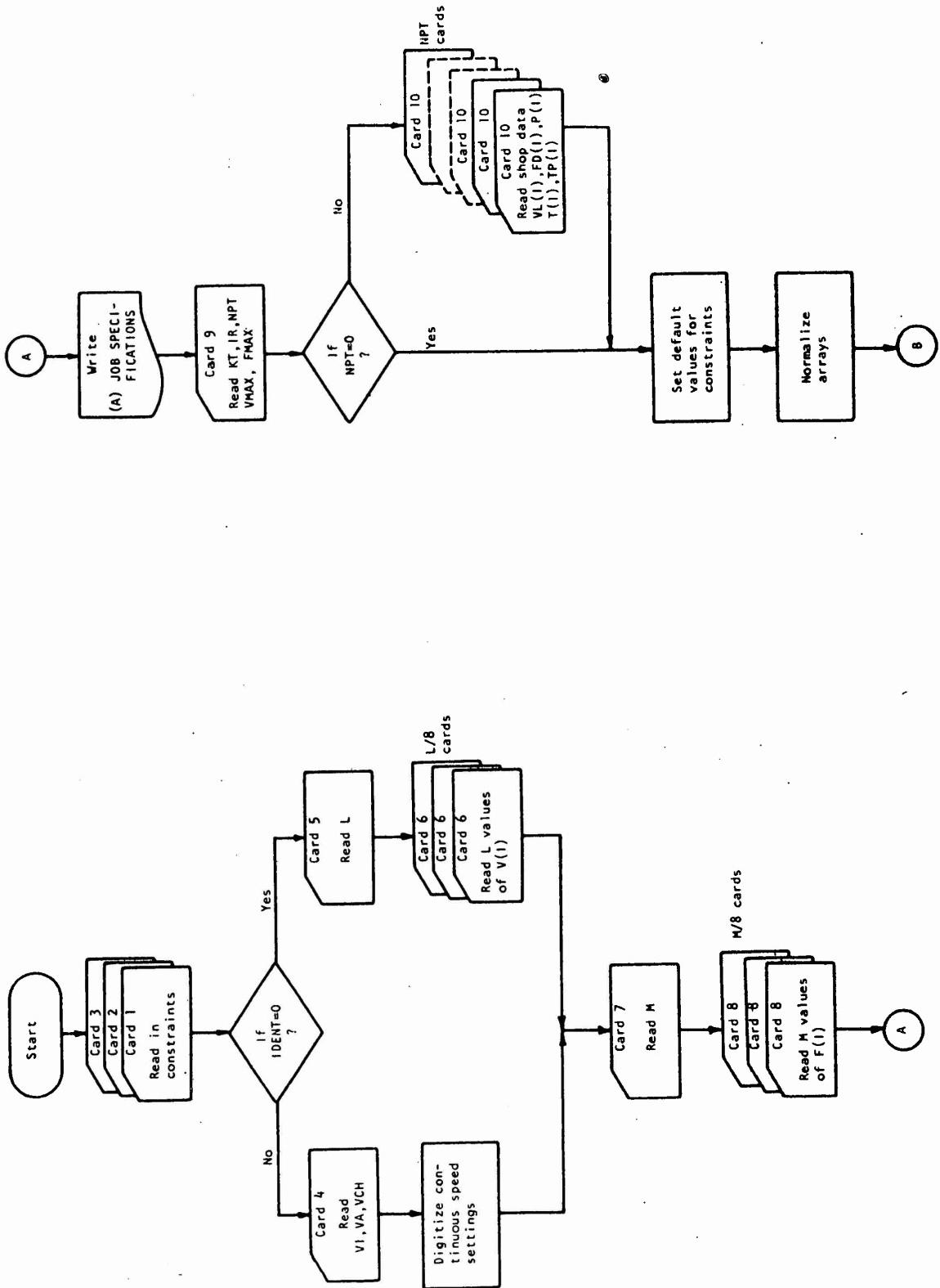
(CONTINUED)
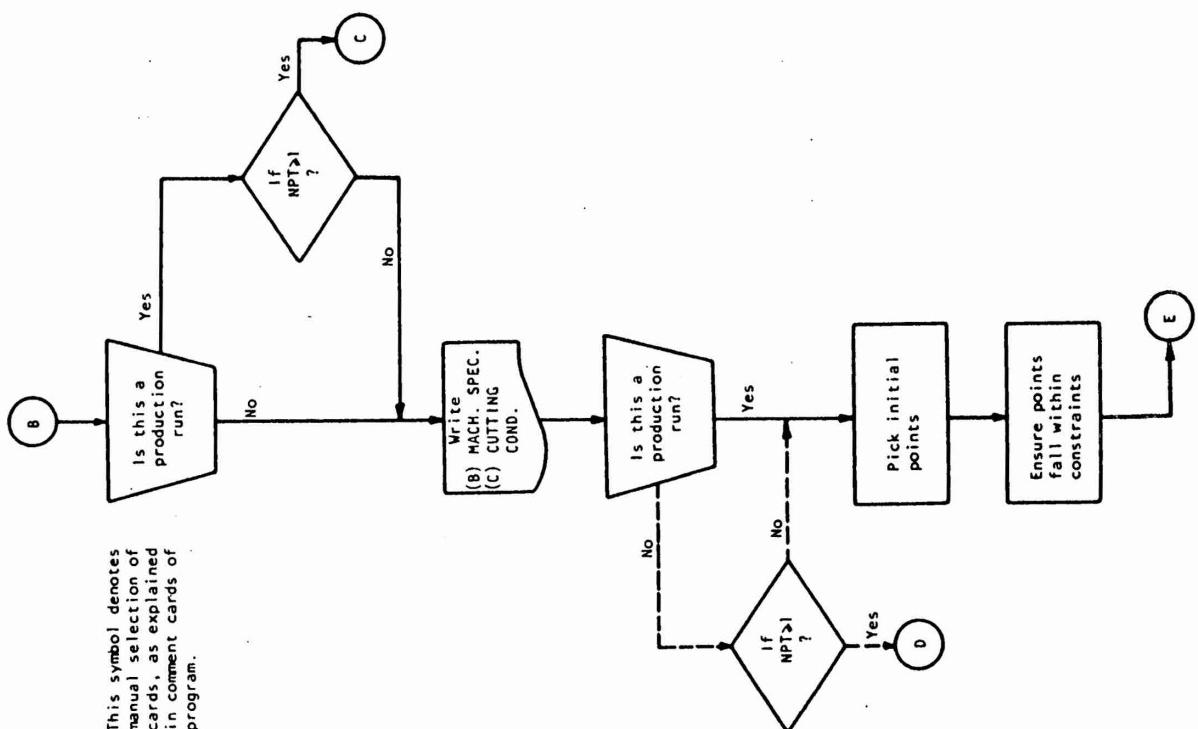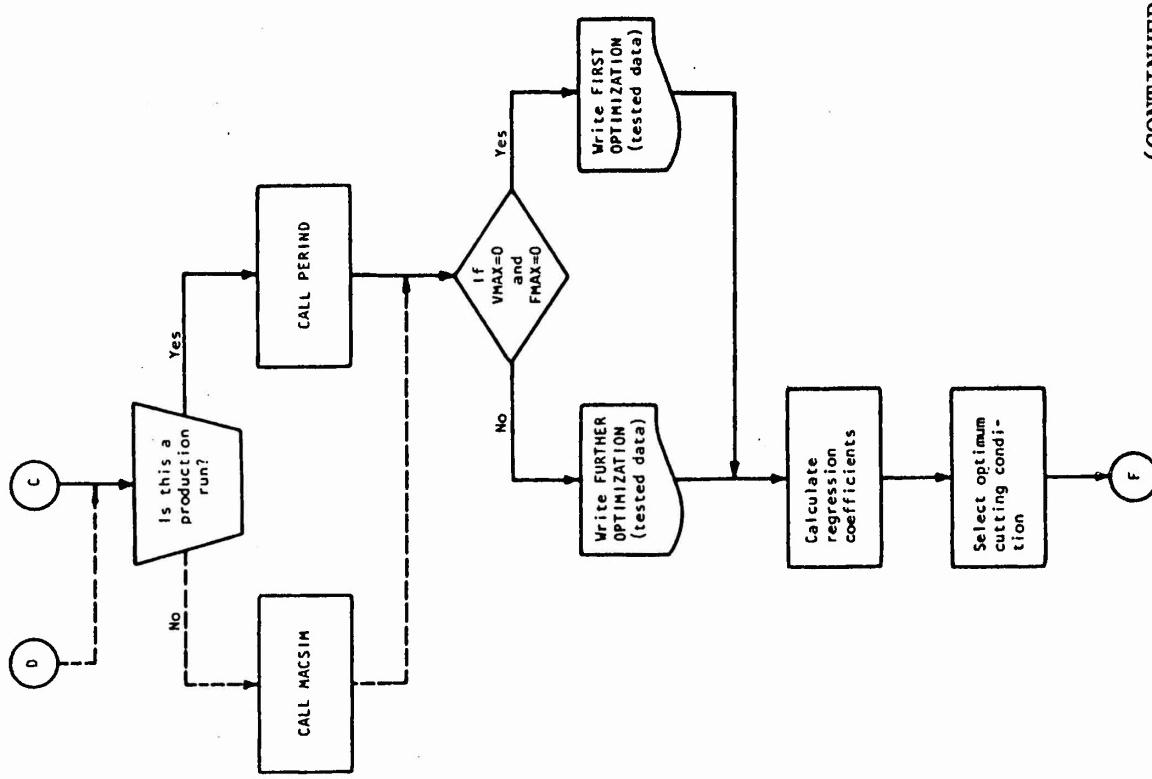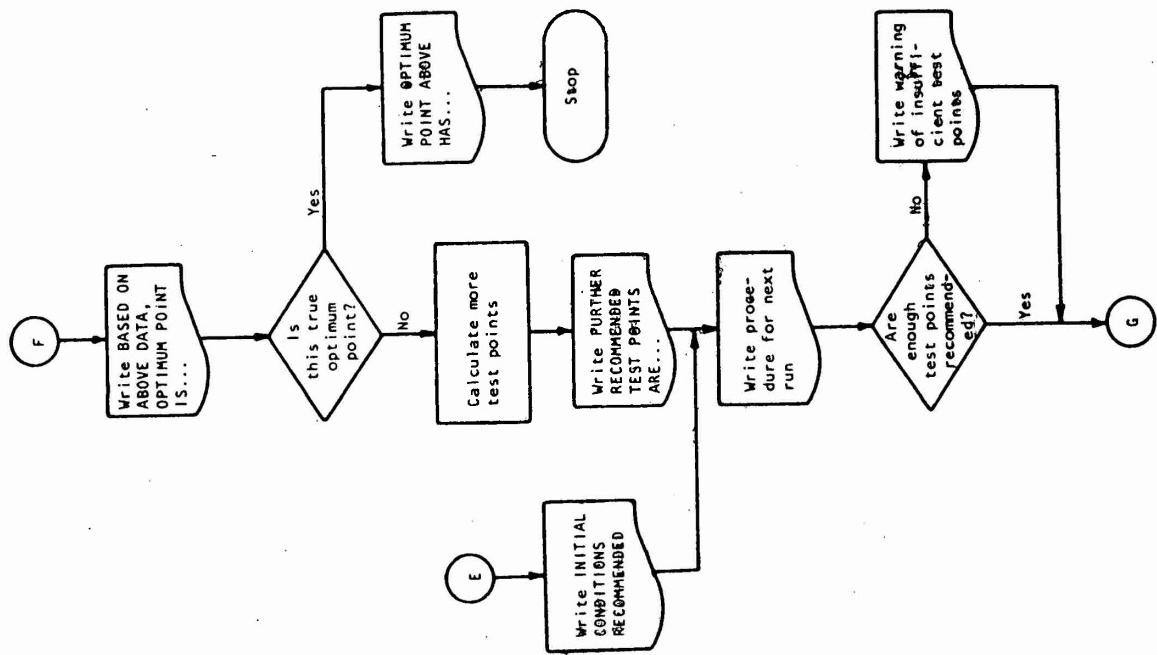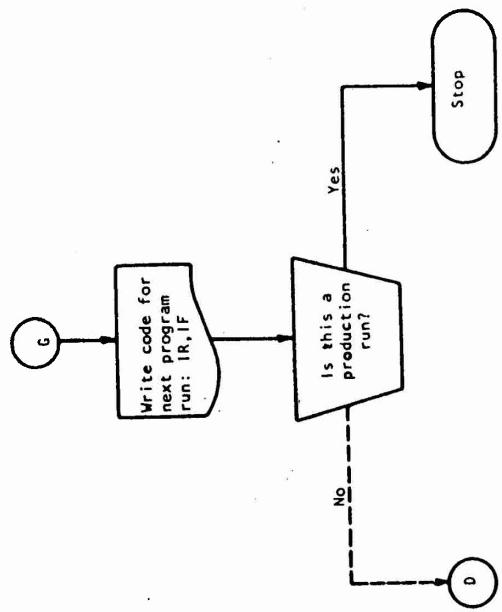
FORMAT GUIDE FOR <u>PIM</u> INPUT DATA CARDS

Card Type	Variable	Column Number	Number Format	Card Type	Variable	Column Number	Number Format
1	IDENT	1	I1	7	M	1-3	I3
	VLOW	11-20	F10.0				
	VHIGH	21-30	F10.0	8	M values of F(I)	1-10 to	F10.0
	FLOW	31-40	F10.0			71-80	F10.0
	FHIGH	41-50	F10.0				
	VFLOW	51-60	F10.0				
	VFHIGH	61-70	F10.0				

Must be ordered from low to high. If M is greater than 8, continue on another card, same columns.

Card Type	Variable	Column Number	Number Format	Card Type	Variable	Column Number	Number Format
2	SHP	1-10	F10.0	9	KT	1-2	I2
	HPM	11-20	F10.0		IR	3-4	I2
	EFF	21-30	F10.0		IF,NPT	5-6	I2
	DC	31-40	F10.0		NOP,VMAX	11-20	F10.0
					FOP,FMAX	21-30	F10.0

Blank card on initial run (all values 0).

Card Type	Variable	Column Number	Number Format	Card Type	Variable	Column Number	Number Format
3	FLO	1-10	F10.0	10	VL(I)	1-10	F10.0
	TLC	11-20	F10.0		FD(I)	11-20	F10.0
	Q	21-30	F10.0		P(I)	21-30	F10.0
	D	31-40	F10.0		T(I)	31-40	F10.0
					TP(I)	41-50	F10.0

Card Type	Variable	Column Number	Number Format
4	VI	1-10	F10.0
	VA	11-20	F10.0
	VCH	21-30	F10.0

Include only if IDENT = 1.

NPT or IF values. Include only when IF $\neq$ 0. When IF is greater than 1, continue on other cards, same columns.

Card Type	Variable	Column Number	Number Format
5	L	1-3	I3

Include only if IDENT = 0.

Card Type	Variable	Column Number	Number Format
6	L values of V(I)	1-10 to	F10.0
		71-80	F10.0

Must be ordered from low to high. If L is greater than 8, continue on another card, same columns.

Format I5: Integer-valued number, no more than 5 digits or 4 digits and minus sign, no decimal point. Right justified. Read blank as 0. Examples:

0	1	2	0	0	read	1200
-	1	2	0	0	read	-1200
		2	0	0	read	200
		2			read	200
			2		read	2
					read	0

Format F10.0: Real-valued number, no more than 9 digits and decimal point. Minus sign may precede. Computer prints decimal point in 10th column if number has none. Read blank as 0. Examples:

			1	0	.					read	10.
						1	0	.		read	10.
	1	3	.	7	3	3				read	13.733
				1	0	0	.			read	100.
										read	0

```
      IMPLICIT REAL * 8 )A-H,O-Z)     THIS IS A LISTING OF THE PIM PROGRAM.
      COMMON HPMC,F,V,FO,VL,PI,NPT
      DIMENSION P)10),T)10),TP)10)
      DIMENSION PR)10),CU)10)
      DIMENSION V)100),F)100),VL)10),FO)10),ISUBV)10),ISUBF)10)
      DIMENSION COF)6)
      DIMENSION PI)10)
C*************************************************************
C          THIS PROGRAM IS DESIGNED TO DETERMINE THE OPTIMUM
C     MACHINING PARAMETERS FOR A GIVEN OPERATION. LEAST SQUARES
C     REGRESSION ANALYSIS IS USED TO LOCATE THE OPTIMUM VALUE OF
C     THE PERFORMANCE INDEX. THE PROGRAM IS USED TO ANALYZE THE
C     RESULTS OF THE TESTS RUN ON THE OPERATION AND TO RECOMMEND
C     FURTHER TESTING WHEN NECESSARY.
C
C          THIS PROGRAM IS NOW WRITTEN TO SIMULATE THE INTERACTION
C     BETWEEN THE USER AND THE PROGRAM WITH THE USE OF A MACHINE
C     SIMULATION SUBROUTINE)MACSIM).
C
C          ALL CARDS WITH A 'C*' IN THE FIRST TWO COLUMNS MUST BE MADE
C     ACTIVE FORTRAN STATEMENTS FOR AN ACTUAL RUN. THIS IS DONE BY
C     DUPLICATING THE CARD LEAVING THE FIRST TWO COLUMNS BLANK.
C
C          ALSO ALL CARDS NOTED TO BE REMOVED MUST BE DONE BEFORE AN ACTUAL
C     RUN
C*************************************************************
    1 FORMAT)I1,9X,6F10.0)
    2 FORMAT)4F10.0)
    3 FORMAT)3I2,4X,2F10.0)
    4 FORMAT)I3)
    5 FORMAT)8F10.0)
    7 FORMAT)5F10.0)
  900 FORMAT)'1',9X,'SYSTEM FOR OPTIMIZING MACHINE PARAMETERS'/
     1'+',9X,42('_')/
     1'0',12X,'METHOD NO.1: OPTIMIZATION OF PERFORMANCE INDEX')'/
     2'+',4X,'(A) JOB SPECIFICATION:'/'+',9X,18)'_')/
     3' ',13X,'OPERATION: ',26('_')/
     3'+',13X,'WORKPIECE: NAME =',10('_'),',  ',NUMBER =',10('_')/
     4' ',24X,'MATERIAL =',10('_'),',  ',HARDNESS =',10('_')/
     5' ',24X,'SIZE: DIAMETER =',6('_'),' IN.',LENGTH =',6('_'),' IN.'/
     6' ',24X,'CUTTING TOOL: TYPE = ',8('_'),' , GRADE = ',8('_')/
     7'+',27X,'GEOMETRY = ',24('_'))
  910 FORMAT)'0',13X,'COST DATA: LABOR AND OVERHEAD RATE)RLO) = $',
     1F4.2,'/MIN.'/
     2'0',24X,'TOOLING COST PER CUTTING EDGE)TLC) = $',F4.2,'/EDGE')
  920 FORMAT)'-',4X,'(B) MACHINE TOOL SPECIFICATIONS:'/'+',9X,28('_')/
  925 FORMAT)'0',11X,'AVAILABLE SPINDLE SPEEDS (IN REV./MIN.) IN',I3,
     1' STEPS:'/'+',11X,24('_')/
  926 FORMAT)'0',11X,'USABLE SPINDLE SPEEDS (IN REV./MIN) IN',I3,
     1' STEPS:'/'+',11X,21('_')/
  930 FORMAT)'-',11X,'AVAILABLE FEEDS (IN INCHES/REV.) IN',I3,' STEPS:'/
     1'+',11X,16('_')/
  931 FORMAT)'0',11X,'USABLE FEEDS (IN INCHES/REV.) IN',I3,' STFPS:'/
     1'+',11X,12('_')/
  935 FORMAT)11X,10F10.1)
  940 FORMAT)11X,10F10.5)
  945 FORMAT)'-',11X,'HORSEPOWER(MAX) ',I3('_'),'(HPM) = ',F4.1,' HP.'/
     1'+',11X,'EFFICIENCY OF MACHINE TOOL --(EFF.) = ',F4.1,'%'/
     2' ',11X,'SPECIFIC HORSEPOWER ----------)SHP) = ',F5.2,' HP/CU-IV/M
     3IN'/
     4'+',11X,'MAX. DEPTH OF CUT EXPECTED --(OCM) = ',F5.2,' IN.'/
  950 FORMAT)'-',4X,'(C) THE CONSTRAINTS FOR CUTTING CONDITION:'/
     1'+',9X,38('_')/
  955 FORMAT)' ',13X,'CUTTING SPEED)FPM) RANGE: ',F6.1,' < V < ',F6.1)
  960 FORMAT)' ',13X,'SPINDLE SPEED)RPM) RANGE: ',F6.1,' < N < ',F6.1)
  965 FORMAT)' ',13X,'FEED(IPR) RANGE:',10X,F6.4,' < F < ',F6.4)
  970 FORMAT)'1',5X,'INITIAL TESTING CONDITIONS RECOMMENDED'/
     1'+',5X,39('_'))
  971 FORMAT)'0',10X,'SPINDLE SPEED(N)     FEED(F)'/
     1'+',10X,16)'_'),4X7('_')/
     2' ',14X,'(RPM)',12X,')IPR)'/)
  975 FORMAT)' ',7X,')I1,8')',3X,F6.1,11X,F6.4)
  976 FORMAT)'-',13X,54('*')/
     114X,'*THE OPTIMUM POINT ABOVE HAS SATISFIED THE STOPPING    *'/
     214X,'*CRITERIA.  NO FURTHER TESTING IS NEEDED.  THE         *'/
     314X,'*ABOVE MACHINING PARAMETERS CAN BE CONSIDERED AS THE   *'/
     414X,'*OPTIMUM CONDITION FOR THE GIVEN MACHINING OPERATION.  *'/
     514X,54('*'))
  980 FORMAT)'-',10X,'PROCEDURE:'/'+',10X,10('_')/
     1'0',7X,'1A) TEST ALL',I2,' CONDITIONS OF THE ABOVE RECOMMENDED'/
     2' ',12X,'COMBINATION OF SPEEDS AND FEEDS AT THE WORKSHOP'/
     3' ',12X,'UNDER THE PRODUCTION CONDITIONS.'/
     4'0',7X,'1B) COLLECT AND RECORD THE FOLLOWING PRODUCTION DATA'/
     5' ',12X,'OBTAINED FROM THE SHOP AFTER THE PRODUCTION TESTS'/
     6' ',12X,'AS SPECIFIED:'/'+',16X,'11) NUMBER OF PARTS PRODUCED)N)'/
     7'/',20X,'DURING A SPECIFIED TIME PERIOD)T)'/
     8'0',16X,'12) NUMBER OF TOOL CHANGES(NTC) MADE'/
     9'0',20X,'DURING A SPECIFIED TIME PERIOD)T)./
     1'0',7X,')C) FEED BACK THE ABOVE )NFORMATION BY PUNCHING'/
     2' ',12X,'INPUT CARDS AS SPECIFIED IN THE WRITE-UP.')
  981 FORMAT)'1',10X,'FURTHER OPTIMIZATION ANALYSIS BY COMPUTER WITH',
     1'THE PRODUCTION'/'+',10X,52)'_')./
     211X,'TEST RESULTS UNDER THE PREVIOUSLY RECOMMENDED TEST CONDITIONS
     3:'/'+',10X,62('_'))
  982 FORMAT)'0','ANALYSIS #',12/'+',+,10('_'))
  983 FORMAT)'-',+,'***WARNING-INSUFFICIENT NUMBER OF POINTS')
  985 FORMAT)'-',5X,'CODES FOR NEXT RUN OF THE COMPUTER PROGRAM:'/
     1'0',19X,'IR = ',I1/' ',19X,'IF = ',I1)
  986 FORMAT)'-',10X,'FURTHER TESTING CONDITIONS RECOMMENDED BASED UPON
     1THE OPTIMUM CONDITION'/'+',5X,'FROM THE PREVIOUS TEST RESULTS.'/
  990 FORMAT)'1',10X,'FIRST OPTIMIZATION ANALYSIS BY COMPUTER WITH'/
     1'+',10X,44('_')./
     2'+',10X,33)'_'))
  991 FORMAT)'//-',10X,'BASED UPON THE ABOVE PRODUCTION TEST RESULTS, T
     1HE FOLLOWING'/
     2' ',6X,'CUTTING CONDITION IS RECOMMENDED AS THE OPTIMUM CONDITION
     3 TO BE'/' ',6X,'TESTED FURTHER:'/
     4' ',19X,34('*')/' ',19X,'*',32X,'*'/
     5' ',19X,'* SPINDLE SPEED(NOP) : ',F6.1,' RPM *'/
     6' ',19X,'* FEED)FOP): ',10X,F6.4,' IPR *'/' ',19X,'*',32X,'*')
  993 FORMAT)' ',19X,34('*'))
  994 FORMAT)'0',19X,'** CUTTING SPEED)VOP) :',F6.1,' FPM **)
  995 FORMAT)'0',19X,'SPINDLE SPEED(N)      FEED(F)    COST INDEX PRODUCTIO
     1N RATE INDEX'/'+',10X,16('_'),' ),3X,10('_'),1,2X,21('_')/
     2' ',14X,'(RPM)',11X,'(IPR)',6X,'(CU',11X,'(PR)'/)
  996 FORMAT)/' THE FOLLOWING DATA IS SIMULATED PRODUCTION DATA'/)
  997 FORMAT)' ',4X,'(',I1,')',6X,F7.1,9X,F7.5,5X,F5.3,11X,F6.2)
```

```fortran
  999 FORMAT(//' THE PERFORMANCE INDEX IS DEFINED AS:'//
     1*(1.-Q)*(1./CU)    WHERE  Q=',F3.1/)
C
      VMAX=C.
      FMAX=C.
      IR=C
      ISTOP=D
      KT=0
      IZ=0
C
C     READ IN CONSTRAINTS AND CONSTANTS.
      READ(5,1)IDENT,VLOW,VHIGH,FLOW,FHIGH,VFLOW,VFHIGH
      READ(5,2) SHP,HPM,EFF,DC
      READ(5,5) RLO,TLC,Q,D
      HPM=(HPM*EFF)/(3.14159*D*DC*SHP)
      IF(IDENT.EQ.0) GO TO 9
C
C     DIGITIZE CONTINUOUS SPEED SETTINGS
C
      READ(5,5) VI,VA,VCH
      XS=(VA-VI)/VCH+1.
      L=XS
      IF(IXS-L).GT.0) L=L+1
      DO 8 I=1,L
      XS=1
    8 V(I)=VI+(IXS-1.)*VCH
      GO TO 10
C
C     READ IN SPEED AND FEED SETTINGS
C
    9 READ(5,4) L
      READ(5,5)(V(I),I=1,L)
   10 READ(5,4) M
      READ(5,5) (F(I),I=1,M)
      WRITE(6,900)
      WRITE(6,910) RLO,TLC
      WRITE(6,925) L
      WRITE(6,935)(V(I),I=1,L )
      WR(TE(6,930) M
      WR(TE(6,940)(F(I),I=1,M )
C
      READ(5,3) KT,IR,NPT,VMAX,FMAX
C
      IF(NPT.NE.0) READ(5,7)(VL(I),FO(I),P(I),T(I),TP(I),I=1,NPT)
C
C     SET DEFAULT VALUES FOR CONSTRAINTS
      IF(VHIGH.EQ.0) VHIGH=V(L)
      IF(VLOW.EQ.0) VLOW=V(1)
      IF(FHIGH.EQ.D) FHIGH=F(M)
      IF(FLOW.EQ.C) FLOW=F(1)
C
C     NORMALIZE ARRAYS CONTAINING SETTINGS- CONSIDER SETTINGS
C     ONLY WITHIN CONSTRAINTS.


      P14=9*P2

      DO 50 I=1,L
      IF(V(I).LT.VLOW) GO TO 50
      IMOVE=I-1
      GO TO 55
   50 CONTINUE
   55 IL=L-(IMOVE+1)
      DO 60 I=1,IL
      V(I)=V(IMOVE+I)
      IF(V(IMOVE+I).GE.VHIGH) GO TO 65
   60 CONTINUE
      L=L
   65 DO 70 I=1,M
      IF(F(I).LT.FLOW) GO TO 70
      IMOVE=I-1
      GO TO 75
   70 CONTINUE
   75 IM=M-(IMOVE+1)
      DO 80 I=1,IM
      F(I)=F(IMOVE+I)
      IF(F(IMOVE+I).GE.FHIGH) GO TO 85
   80 CONTINUE
      IF(IMOVE.EQ.D) I=M
   85 M=I
C*****************************************************************
C     NPT SHOULD BE ZERO ZERO ONLY ON THE FIRST INTIAL RUN, ASSUMING
C     NO INTIAL POINTS ARE SUPPLIED BY THE USER.
C*    IF(NPT.NE.0) GO TO 300
C*****************************************************************

      WRITE(6,920)
      XEFF=10*.*EFF
      WRITE(6,945) HPM,XEFF,SHP,DC
      WRITE(6,950)
      IF(VFLOW.NE.0..AND.VFHIG4.NE.0.) WRITE(5,955) VFLOW,VFHIGH
      WRITE(6,960) VLOW,VHIG4
      WRITE(6,965) FLOW,FHIGH
      WRITE(6,926)L
      WRITE(6,935)(V(I),I=1,L)
      WRITE(6,931)M
      WRITE(6,940)(F(I),I=1,M)
C*****************************************************************
C     CARDS IN THIS BLOCK SHOULD BE REMOVED FOR ACTUAL RUN
C     IF(NPT.NE.0) GO TO 500
C*****************************************************************
C
C     UP TO 9 INTIAL POINTS ARE PICKED
C
C     THE 9 INITIAL POINTS ARE PICKED IN THE FOLLOWING MANNER:
C     1) THREE POINTS ARE PICKED ALONG A ROW WHICH BISECTS THE AREA
C        GIVEN BY THE FEED AND SPEED CONSTRAINTS.  THESE POINTS DIVIDE
C        THE ROW(BOTTOM ROW) INTO EQUAL LINE SEGMENTS.  IF ANY OF
C        THESE POINTS EXCEEDS THE HPM CONSTRAINT THEY ARE MOVED
C        HORIZONTALLY TO THE LEFT UNTIL THEY ARE WITHIN THE
```

61

```
C      CONSTRAINT(USING PTMOVE).
C      2) FOR EACH POINT PICKED ON THE BOTTOM ROW A COLUMN IS
C      ESTABLISHED BY PICKING TWO ADDITIONAL POINTS, ONE ON THE
C      FHIGH CONSTRAINT AND THEDTHER MIDWAY BETWEEN THE ORIGINAL
C      POINT AND CONSTRAINT.
C      3)  IF THE HPM CONSTRAINT EXCLUDES ANY OF THE ADDITIONAL
C      POINTS, THEY ARE MOVED VERTICALLY DOWN TO WITHIN THIS
C      CONSTRAINT(PTMOVE).
C      4)  HENCE IF THE HP4 CONSTRAINT DID NOT EFFECT ANY POINTS,
C      THESE 9 INITIAL POINTS WOULD FROM A 3 BY 3 POINT SQUARF.
C      THE POINTS ARF STOPED AS FEED(FD) AND SPEED(VL) SETTINGS.
      NPT=0
C      NPT IS THE NUMBER OF POINTS PICKED
      AK=.25
      IY=M/2
C
C      PICK BOTTOM ROW POINTS
C
      DO 100 I=1,3
      IX=AK*L
      CALL PTMOVE(IX,IY,F,V,0)
      IF (I.EQ.1) GO TO 90
C
C      CHECK IF POINT IS UNIQUE
C
      IF(V(IX).EQ.VL(II-1)) GO TO 150
   90 NPT=NPT+1
      FD(NPT)=F(IY)
      VL(NPT)=V(IX)
      ISUBV(NPT)=IX
      ISURF(NPT)=IY
  100 AK=AK+.25
C
C      PICK TOP ROW POINTS
C
  150 K=NPT
      DO 250 I=1,K
      IX=ISUBV(I)
      IY=M
      CALL PTMOVE (IX,IY,F,V,1)
C      CHECK IF POINT IS UNIQUE
      IF(F(IY).EQ.FD(I)) GO TO 250
      NPT=NPT+1
      FD(NPT)=F(IY)
      VL(NPT)=V(IY)
      ISURV(NPT)=IX
      ISURF(NPT)=IY
C
C      PICK MIDDLE ROW POINTS
C
      XS=(IY+ISURF(I))/2.
      IY=XS
      IF(XS-IY.GT.0) IY=IY+1
C
C      CHECK IF POINT IS UNIQUE
C      IF(F(IY).EQ.FD(NPT)) GO TO 250
      NPT=NPT+1
      FD(NPT)=F(IY)
      VL(NPT)=V(IX)

      ISUBV(NPT)=IX
      ISURF(NPT)=IY
  250 CONTINUE
C
      WRITE(6,970)
      WRITE(6,971)
  251 DO 249 I=1,NPT
  249 WRITE(6,975) I,VL(I),FD(I)
      WRITE(6,980) NPT
      IF(NPT.LT.6.AND.IR.EQ.0).OR.(NPT.LT.3.AND.IR.EQ.1))WRITE(6,983)
      WRITE(6,985) IR,NPT
C
C****************************************************************
C*      GO TO 715
C****************************************************************
C
C*300 CALL PERIND(P,T,TP,PR,CU,Q,RLO,TLC)
C****************************************************************
C
C****************************************************************
C      CARDS IN THIS BLOCK SHOULD BE REMOVED FOR ACTUAL RUN
  500 CALL MACSIMICU,PR,Q,I7,RLO,TLC,KT)
C****************************************************************
C
      IF(VMAX.EQ.0..AND.FMAX.EQ.0.) GO TO 490
      WRITE(6,981)
      WRITE(6,982) KT
      IF(IZ.EQ.0) GO TO 495
      WRITF(6,996)
      GO TO 495
  490 WRITE(6,990) NPT
      IF(IZ.EQ.0) GO TO 495
      WRITE(6,999)
      WRITE(6,999) Q
  495 WRITE(6,995)
      WRITE(6,997)II,VL(I),FD(I),CU(I),PP(I),I=1,NPT)
C
C      DETERMIN REGRESSION COEFFICIENTS
      IF(IR.EQ.0) GO TO 520
      CALL PGRS2(COF)
      GO TO 525
  520 CALL RGRS1(COF)
C
C      WITH REGRESSION COEFFICIENTS COMPUTED, SCAN ALL POSSIBLE
C      SETTINGS FOR THE MAXIMUM PI.
C      POINTS ARE SCANNED HORIZONTALLY, LEFT TO RIGHT, FROM BOTTOM TO
C      TOP.  IF THE OLD FMAX IS ON THE FHIGH LINE(WHFN IR=1) ONLY THE
C      FHIGH IS SEARCHED
  525 PIMAX=0.
      DO 560 J=1,M
      IF(IR.EQ.0) GO TO 527
      IF(J.NE.M) GO TO 560
```

62

```
527 DO 550 I=1,L
    IY=J
    IF(IR.EQ.0) GO TO 529
    CALL PLMOVE(I,IY,F,V,IC)
    IF(IC.NE.1) GO TO 550
529 IF(HPMCK(IV(I),F(IJ)).NE.0) GO TO 550
    IF(IR.EQ.1) GO TO 528
    FPI=FUNP1(I,IY,COF)
    GO TO 530
528 FPI=FUNP2I1,COF)
530 IF(FPI.LE.PIMAX) GO TO 550
    PIMAX=FPI
    IB=I
    JB=IY
550 CONTINUE
560 CONTINUE
C
C   ONCE PIMAX IS FOUND ADDITIONAL POINTS MUST BE PICKED
    IF(IFIJB).EQ.FMAX.AND.VI(IB).EQ.VMAX) GO TO 700
    FMAX=F(JB)
    VMAX=V(IB)
    CALL PICK(FMAX,VMAX,L,M,FHIGH,IR,JB,IB)
C
    WRITE(6,986)
    WRITE(6,971)
C
C**********************************************************
C   CARDS IN THIS BLOCK SHOULD BE REMOVED FOR ACTUAL RUN
    KT=KT+1
    IF(KT.GT.10) STOP
C**********************************************************
    GO TO 251
C
700 WRITE(6,976)
715 WRITE(6,720)
720 FORMAT('1','***END OF RUN')
    STOP
    END
    SUBROUTINE PTMOVE(IX,IY,F,V,IC)
C
C   THIS ROUTINE MOVES A POINT TO WITHIN THE HPM CONSTRAINT
C   IF NECESSARY.
C   IC - DETERMINES WHAT DIRECTION THE POINT IS TO BE MOVED
C        IC = 0 - MOVES POINT HORIZONTALLY
C        IC = 1 - MOVES POINT VERTICALLY
C
    IMPLICIT REAL * 8 (A-H,O-Z)
    DIMENSION V(100),F(100)
    K=0
50  IF(HPMCK(V(IX),F(IY)).EQ.0) RETURN
    IF(IC.EQ.0) GO TO 100
    IY=IY-1
    IF(IY.LE.0) GO TO 200
    GO TO 50
100 IX=IX-1
    IF(IX.LE.0) GO TO 200
    GO TO 50
C
C   PLMOVE TRIES TO MOVE THE POINT, OTHERWISE SETS A RETURN CODE
    ENTRY PLMOVE(IX,IY,F,V,IC)
    K=1
    IC=1
    GO TO 50
200 IF(IK.EQ.0) GO TO 250
    IC=0
    RETURN
250 WRITE(6,300)
300 FORMAT('0***WARNING-HPM CONSTRAINT EXCLUDES POSSIBLE SETTINGS')
    STOP
    END
    FUNCTION HPMCK(X,Y)
C
C   THIS ROUTINE CHECKS IF A POINT (X,Y) IS WITHIN THE
C   HPM CONSTRAINT.
C     A RETURN OF 0 INDICATES THE POINT IS WITHIN THE CONSTRAINT
C     A RETURN OF 1 INDICATES IT IS NOT WITHIN THE CONSTRAINT.
C
    IMPLICIT REAL * 8 (A-H,O-Z)
    COMMON HPMC,F,V,FO,VL,PI,NPT
    DIMENSION F(100),V(100),FD(10),VL(10),PI(10)
    HPMCK=0
    YSTAR=HPMC/X
    IF(Y.LE.YSTAR) RETURN
    HPMCK=1
    RETURN
    END
    SUBROUTINE MACSIM(ICU,PR,Q,IZ,PLO,TLC,KT)
C
C   THIS SIMULATION SUBROUTINE IS USED ONLY DURING DEVELOPMENT PHASES
C   OF THE PROJECT.  ONCE COMPLETED AND THE PROGRAM IS USED INTER-
C   ACTIVELY WITH A USER, THE ROUTINE IS NO LONGER NEEDED.
C
    IMPLICIT REAL * 8 (A-H,O-Z)
    REAL*4 ERR
    REAL * 8 RAND
    COMMON HPMC,F,V,FO,VL,PI,NPT
    DIMENSION PI(10),FD(10),VL(10)
    DIMENSION F(100),V(100)
    DIMENSION CU(10),PR(10)
    DATA A,BB,C,TCT,CO,CE,TM,O,XL/.203.,194.270.,.1.,.2,.5.,.6.,24./
890 FORMAT('-',20X,'COMPUTER SIMULATION CONDITIONS'/'+',20X,30('.')'/
   1'0',30X,'D = ',F5.2/31X,'L = ',F5.2/3?X,'TM = ',F5.2/29X,'TCT = ',
   2,F5.2/29X,'RLO = ',F5.2/29X,'TLC = ',F5.2/'9',24X,'% ERROR = ',
   3F4.0,'%'/20X,'V*T**A*F**B*DC**G=C',/20X,'A = ',F5.3/20X,'B = ',F5.
   43/20X,' G = 0.'/20X,'C = ',F5.1)
    DO 100 I=1,NPT
    TC=XL/IFD(I)*VL(I))
    VFPM=VL(I)*D*3.14159/12.
    T=(C/(VFPM*FD(I)**BB))**II./A)
    X=CO*TM*CO*TC+(CO*TCT*TC)/T+ICE*TC/T
C   RANDOM ERROR IS INDUCED
    ERR=.10
    K=RAND(2.)*1.
```

```fortran
      X=(RAND(ERR)*(-1.)**K)*X+X
      CU(I)=X
      X=1./(T4*TC+TC1*TC/T)
C     RANDOM ERROR IS INDUCED
      K=RAND(2,)+1
      X=(RAND(ERR)*(-1.)**K)*X+X
      PR(I)=X
      IZ=1
  100 PI(I)=Q*PR(I)+(1.-Q)/CU(I)
      ERR=ERR*100.
      IF(KT.EQ.0) WRITE(6,890) O ,XL,T4,TCT,RLO,TLC,ERR,A,BB,C
      RETURN
      END
      SUBROUTINE RGRS1(COF)
C
C     THIS ROUTINE PERFORMS A LEAST SQUARES FIT REGRESSION ON THE
C     MODEL; PI = B0 + B1V + B2F + B3V**2 + B4F**2 + B5VF
C     NPT - SAMPLE SIZE
C     FD AND VL - INDEPENDANT VARIABLES
C     PI - DEPENDANT VARIBLE
C     COF - WILL CONTAIN THE REGRESSION COEFFECIENTS UPON COMPLETION
C     OF THE ROUTINE.
C
C     TO CHANGE THIS ROUTINE TO FIT A DIFFERENT MODEL, SIMPLY;
C     1) CHANGE DO-PARAMETERS FROM 6 TO THE NUMBER OF COEFFICIENTS
C        TO BE ESTIMATED.
C     2) CHANGE TALLY ROUTINE TO SUIT EACH TERM OF THE NEW MODEL.
      IMPLICIT REAL * 8 (A-H,O-Z)
      COMMON HPMC,F,V,FD,VL,PI,NPT
      DIMENSION F(100),V(100)
      DIMENSION FD(10),VL(10),PI(10),COF(6),ARAY(6,6)
C     INITIALIZE ARRAYS TO ZERO
      DO 50 J=1,6
   50 COF(J)=0.
      DO 75 I=1,NPT
      Y=PI(I)
      V1=VL(I)
      F1=FD(I)
      COF(1)=COF(1)+Y
      COF(2)=COF(2)+Y*V1
      COF(3)=COF(3)+Y*F1
      COF(4)=COF(4)+Y*V1*V1
      COF(5)=COF(5)+Y*F1*F1
   75 COF(6)=COF(6)+Y*V1*F1
C     COMPUTE SUMMATION MATRIX
      DO 100 I=1,6
      DO 100 K=1,6
      ARAY(I,K)=0.
      DO 100 J=1,NPT
      X=TALLY(I,J)*TALLY(K,J)
  100 ARAY(I,K)=ARAY(I,K)+X
C     SOLVE LINEAR SYSTEM OF EQUATIONS
      CALL DLEQD(ARAY,COF,6,1,6,6,DET)
      RETURN
      END
      FUNCTION TALLY(IK,J)
C
C     THIS ROUTINE IS USED WITH THE REGRESSION ROUTINES.
C     EACH "GO TO " RELATES TO A TERM IN THE MODEL.
      IMPLICIT REAL * 8 (A-H,O-Z)
      COMMON HPMC,F,V,FD,VL,PI,NPT
      DIMENSION F(100),V(100),FD(10),VL(10),PI(10)
      GO TO (10,20,30,40,50,60),IK
   10 TALLY=1.
      GO TO 100
   20 TALLY=VL(J)
      GO TO 100
   30 TALLY=FD(J)
      GO TO 100
   40 TALLY=VL(J)*VL(J)
      GO TO 100
   50 TALLY=FD(J)*FD(J)
      GO TO 100
   60 TALLY=VL(J)*FD(J)
  100 RETURN
      END
      SUBROUTINE RGRS2(COF)
C
C     USED FOR THE MODEL; PI = B0 + B1V + B2V**2
C
      IMPLICIT REAL * 8 (A-H,O-Z)
      COMMON HPMC,F,V,FD,VL,PI,NPT
      DIMENSION F(100),V(100),FD(10),VL(10),PI(10)
      DIMENSION COF(3),ARAY(3,3)
      DO 50 I=1,3
   50 COF(I)=0.
      DO 75 I=1,NPT
      COF(1)=COF(1)+PI(I)
      COF(2)=COF(2)+PI(I)*VL(I)
   75 COF(3)=COF(3)+PI(I)*VL(I)*VL(I)
      DO 100 I=1,3
      DO 100 K=1,3
      ARAY(I,K)=0.
      DO 100 J=1,NPT
      (J)=I
      KJ=K
      IF(I.EQ.3) IJ=4
      IF(K.EQ.3) KJ=4
      X=TALLY(IJ,J)*TALLY(KJ,J)
  100 ARAY(I,K)=ARAY(I,K)+X
      CALL DLEQD(ARAY,COF,3,1,3,3,DET)
      RETURN
      END
      FUNCTION FUNPI(I,J,COF)
C
C     EVALUATES   PI=B0 + B1V + B2F + B3 ETC.
C
      IMPLICIT REAL * 8 (A-H,O-Z)
      DIMENSION F(100),V(100),FD(10),VL(10),PI(10)
      COMMON HPMC,F,V,FD,VL,PI,NPT
      DIMENSION COF(6)
      X=V(1)*(COF(2)+V(1)*COF(4))
      X1=F(J)*(COF(3)+F(J)*COF(5))
      FUNPI=COF(1)+X+X1+COF(6)*V(I)*F(J)
      RETURN
      ENTRY FUNP2(I,COF)
```

```
      FUNP2=COF(1)+COF(2)*V(I1)+COF(3)*V(I1)*V(I1)
      RETURN
      END
      SUBROUTINE PICK(FMAX,VMAX,L,M,FHIGH,IR,MB,LB)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON HPMC,F,V,FD,VL,PI,NPT
      DIMENSION V(100),F(100),VL(10),FD(10)
      DIMENSION PI(10)
C
      L4=L/4
      M4=M/4
      IF(L4.LE.1) L4=2
      IF(M4.LE.1) M4=2
      IF(FMAX.EQ.FHIGH) GO TO 530
      IF(HPMCK(VMAX,F(MB+1)).NE.0) GO TO 500
C
C 9 POINTS ARE PICKED CENTERED AROUND VMAX,FMAX
      NPT=1
      DO 100 I=1,3
      IX=LR
      IF(I.EQ.1) GO TO 50
      IX=LB+((I-1.)**I*L4)
      IF(IX.GT.L) IX=L
      IF(IX.LT.1) IX=1
      CALL PTMOVE(IX,MB,F,V,0)
50    DO 100 J=1,3
      IY=MB
      IF(J.EQ.1.AND.I.EQ.1) GO TO 75
      IY=MB+((-1.)**J*M4)
      IF(IY.LT.1) IY=1
      IF(IY.GT.M) IY=M
      CALL PTMOVE(IX,IY,F,V,1)
      IF(ICHECK(IX,IY).NE.0) GO TO 100
75    NPT=NPT+1
      FD(NPT)=F(IY)
      VL(NPT)=V(IX)
100   CONTINUE
      IR=0
      RETURN
C
C 5 POINTS ARE PICKED ALONG THE FHIGH LINE-WITH VMAX IN THE MIDDLE
500   IX=LR
      IY=MB
      CALL PTMOVE(IX,IY,F,V,1)
      NPT=1
      FD(NPT)=F(IY)
      VL(NPT)=V(IX)
      DO 600 I=1,4
      IY=B
      IF(I.GT.2) GO TO 550
      IX=LB+((-1.)**I*L4)
      IF(IX.GT.L) IX=L
      IF(IX.LT.1) IX=1
      GO TO 575
550   IX=LB+((-1.)**I*L4)*2
      IF(IX.GT.L) IX=L
      IF(IX.LT.1) IX=1
575   CALL PLMOVE(IX,IY,F,V,IC)
      IF(IC.EQ.0) GO TO 600
      IF(I.EQ.1) GO TO 576

576   NPT=NPT+1
      FD(NPT)=F(IY)
      VL(NPT)=V(IX)
600   CONTINUE
      IR=1
      RETURN
      END
      FUNCTION ICHECK(IX,IY)
C THIS ROUTINE CHECKS FOR NON-UNIQUE POINTS.   A ZERO IS RETURNED.
C IF POINT IN QUESTION IS UNIQUE, OTHERWISE A ONE IS RETURNED.
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON HPMC,F,V,FD,VL,PI,NPT
      DIMENSION V(100),F(100),VL(10),FD(10),PI(10)
      ICHECK=1
      DO 100 I=1,NPT
      IF(FD(I).EQ.F(IY).AND.VL(I).EQ.V(IX)) RETURN
100   CONTINUE
      ICHECK=0
      RETURN
      END
      SUBROUTINE PERIOD(P,T,TP,PR,CU,Q,RLD,TLC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION V(100),F(100),VL(10),FD(10),PI(10),TI(10),TP(10)
      DIMENSION PR(10),CU(10)
      COMMON HPMC,F,V,FD,VL,PI,NPT
C
      DO 100 I=1,NPT
      PR(I)=P(I)/T(I)
      CU(I)=(RLD*T(I)+TLC*TP(I))/P(I)
100   PI(I)=(1.-Q)*PR(I)+(Q/CU(I))
      RETURN
      END
      SUBROUTINE EDLEQO(A,B,N,M,NDA,NDB,DET)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NDA,NDA),B(NDA,M)
      IF(M.GT.1)GOTO10
      LB=N
      GOTO100
10    LB=NDB
100   CALL MATRX3(A,B,N,M,NDA,NDB,DET,LR)
      RETURN
      END
      SUBROUTINE MATRX3(A,B,N,M,NDA,NDB,DET,LR)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C SUBROUTINE MATRX3 PERFORMS UP TO 3 SIMULTANEOUS OPERATIONS ON A
C      SYSTEM OF MATRICES OF THE FORM  AX = B .
C      THE OPERATIONS ARE...
C      1.) FINDING DET(A)
C      2.) FINDING A-INVERSE.
C      3.) OPTION OF SOLVING FOR X.
C      OPERATIONS 1.) AND 2.) ARE ALWAYS PERFORMED.
C      OPERATION 3.), SOLVING THE SYSTEM FOR ONE OR MORE COLUMNS OF
C      UNKNOWNS, IS ONLY DONE WHEN CALLED FOR BY CONTROL(IE. THE
C      VALUE OF M).
C
C      DEFINITION OF ARGUMENTS...
C      A   IS THE COEFFICIENT MATRIX.
      DIMENSION A(NDA,N),B(LB,M)
```

```
C  B     IS THE CONSTANT MATRIX, OR A DUMMY VARIABLE, CONSTANT,
C        OR ARRAY IF M=0
C  N     IS THE ORDER OF A OR THE NO. OF EQUATIONS AND UNKNOWNS.
C  M     IS THE NUMBER OF SETS OF EQUATIONS OR THE NO. OF COLUMNS
C        IN B.
C        (IF M = 0 OPERATION 3.) IS NOT PUT INTO EFFECT.)
C  NDA   IS THE DIMENSION OF THE FIRST SUBSCRIPT OF A.
C  NDB   IS THE DIMENSION OF THE FIRST SUBSCRIPT OF B.
C        (NDB = 1 IF B IS A VECTOR.)
C  DET IS RETURNED AS THE DETERMINENT OF A.
C  A IS RETURNED AS A-INVERSE.
C  B IS RETURNED AS THE SOLUTION MATRIX IF M IS .NE. 0  .
C
C  METHOD... GAUSS-JORDAN ELIMINATION WITH COMPLETE PIVOTING.
C
C  LIMITATIONS...
C  THE NUMBER OF EQUATIONS MAY NOT EXCEED 100.
C
C  H. D. KNDRLE - MARCH, 1963.  PSU COMPUTATION CENTER.
C
      DIMENSION ISWTCH(100),JROW(100),JCOL(100)
C
C        INITIALIZE.
C
      DET=1.00
      DO 15 J=1,N
15    ISWTCH(J)=0
      DO 215 I=1,N
C
C        SEARCH FOR MAXIMUM PIVOT ELEMENT.
C
      PIVMAX=0.
      DO 65 J=1,N
      IF (ISWTCH(J)) 25,25,65
25    DO 55 K=1,N
      IF (ISWTCH(K)) 35,35,55
35    IF ( OABS(PIVMAX)- OABS(A(J,K))) 45,55,55
45    IROW=J
      ICOL=K
      PIVMAX=A(J,K)
55    CONTINUE
65    CONTINUE
      ISWTCH(ICOL)=1
C
C     IF PIVOT ELEMENT IS NOT ON THE DIAGONAL, SWITCH ROWS TO
C     PUT IT THERE.
C
      IF (IROW-ICOL) 75,115,75
75    DET=-DET
      DO 85 L=1,N
      SAVE=A(IROW,L)
      A(IROW,L)=A(ICOL,L)
85    A(ICOL,L)=SAVE
C
C     CHECK THE STATUS OF M AND REARRANGE B'S ACCORDING TO
C     PREVIOUS PIVOT IF THE MATRIX EQUATION IS TO BE SOLVED.
C
      IF (M) 115,115,95
95    DO 105 L=1,M
      SAVE=B(IROW,L)
      B(IROW,L)=B(ICOL,L)
105   B(ICOL,L)=SAVE
115   JROW(I)=IROW
      JCOL(I)=ICOL
      PIVEL=A(ICOL,ICOL)
      DET=DET*PIVEL
C
C        DIVIDE PIVOT ELEMENT INTO PIVOT ROW.
C
      A(ICOL,ICOL)=1.0
      DO 135 L=1,N
135   A(ICOL,L)=A(ICOL,L)/PIVEL
      IF (M) 165,165,145
145   DO 155 L=1,M
155   B(ICOL,L)=B(ICOL,L)/PIVEL
C
C        OPERATE ON NON-PIVOT ROWS.
C
165   DO 215 J=1,N
      IF (J-ICOL) 175,215,175
175   PIVMAX=A(J,ICOL)
      A(J,ICOL)=0.
      DO 185 L=1,N
185   A(J,L)=A(J,L)-A(ICOL,L)*PIVMAX
C185  A(J,L)=A(J,L)-A(ICOL,L)*PIVMAX
      IF (M) 215,215,195
195   DO 205 L=1,M
205   B(J,L)=B(J,L)-B(ICOL,L)*PIVMAX
215   CONTINUE
C
      RETURN
      END
      FUNCTION RAND(X)
C  STATEMENTS 4,5, AND 6 CHANGED TO AGREE WITH SCIENTIFIC SUR.
C  BY WILLIAM H VERITY ON JUNE 5,1967.
      DATA IX/123456789/
      IY = IX*65539
4     IF(IY)5,6,6
5     IY = IY+2147483647 + 1
6     YFL = IY
      RAND = YFL*0.46566E-9**X
      IX = IY
      RETURN
      ENTRY SAVBAS(ISAVI)
      ISAV = IX
      SAVBAS = 0.0
      RETURN
      ENTRY SETBAS(ISET)
      IX = ISET
      SETBAS = 0.0
      RETURN
      END
```

SYSTEM FOR OPTIMIZING MACHINE PARAMETERS

(A) (METHOD NO. 1: OPTIMIZATION OF PERFORMANCE INDEX)

JOB SPECIFICATIONS:
OPERATION:
WORKPIECE: NAME = _____ , NUMBER = _____
            MATERIAL = _____ , HARDNESS = _____
            SIZE: DIAMETER = _____ IN., LENGTH = _____ IN.
CUTTING TOOL: TYPE = _____ , GRADE = _____
              GEOMETRY = _____

COST DATA: LABOR AND OVERHEAD RATE (RLO) = $0.10/MIN
           TOOLING COST PER CUTTING EDGE (TLC) = $0.20/EDGE

AVAILABLE SPINDLE SPEEDS (IN REV/MIN) IN 50 STEPS:

20.0	40.0	60.0	80.0	100.0	120.0	140.0	160.0	180.0	200.0
220.0	240.0	260.0	280.0	300.0	320.0	340.0	360.0	380.0	400.0
420.0	440.0	460.0	480.0	500.0	520.0	540.0	560.0	580.0	600.0
620.0	640.0	660.0	680.0	700.0	720.0	740.0	760.0	780.0	800.0
820.0	840.0	860.0	880.0	900.0	920.0	940.0	960.0	980.0	1000.0

AVAILABLE FEEDS (IN INCHES/REV) IN 24 STEPS:

0.00110	0.00150	0.00180	0.00240	0.00300	0.00360	0.00420	0.00460	0.00510	0.00560
0.00600	0.00680	0.00780	0.00840	0.00920	0.00940	0.01020	0.01120	0.01200	0.01280
0.01360	0.01470	0.01560	0.01680

(B) MACHINE TOOL SPECIFICATIONS:
HORSEPOWER(MAX) --------------(HPM) = 7.5 HP
EFFICIENCY OF MACHINE TOOL --(EFF) = 60.0 %
SPECIFIC HORSEPOWER ---------(SHP) = 0.75 HP/CU IN./MIN
MAX. DEPTH OF CUT EXPECTED --(DCM) = 0.10 IN.

(C) THE CONSTRAINTS FOR CUTTING CONDITION:
CUTTING SPEED (FPM) RANGE: 200.0 < V < 500.0
SPINDLE SPEED (RPM) RANGE: 127.0 < N < 320.0
FEED (IPR) RANGE: 0.0051 < F < 0.0102

USABLE SPINDLE SPEEDS (IN REV/MIN) IN 10 STEPS:

140.0	160.0	180.0	200.0	220.0	240.0	260.0	280.0	300.0	320.0

USABLE FEEDS (IN INCHES/REV) IN 9 STEPS:

0.00510	0.00560	0.00600	0.00680	0.00780	0.00840	0.00920	0.00940	0.01020

(CONTINUED)

67

INITIAL TESTING CONDITIONS RECOMMENDED:

	SPINDLE SPEED (N) (RPM)	FEED (F) (IPR)
(1)	160.0	0.0068
(2)	220.0	0.0068
(3)	260.0	0.0068
(4)	160.0	0.0102
(5)	160.0	0.0092
(6)	220.0	0.0102
(7)	220.0	0.0092
(8)	260.0	0.0102
(9)	260.0	0.0092

PROCEDURE:

(A) TEST ALL 9 CONDITIONS OF THE ABOVE RECOMMENDED COMBINATION
OF SPEEDS AND FEEDS AT THE WORKSHOP UNDER THE PRODUCTION
CONDITIONS.

(B) COLLECT AND RECORD THE FOLLOWING PRODUCTION DATA OBTAINED
FROM THE SHOP AFTER THE PRODUCTION TESTS AS SPECIFIED:

    (1) NUMBER OF PARTS PRODUCED (NP)
        DURING A SPECIFIED TIME PERIOD (T)

    (2) NUMBER OF TOOL CHANGES (NTC) MADE
        DURING A SPECIFIED TIME PERIOD (T)

(C) FEED BACK THE ABOVE INFORMATION BY PUNCHING INPUT CARDS
AS SPECIFIED IN THE WRITE-UP.

CODES FOR NEXT RUN OF THE COMPUTER PROGRAM:

IR = 0
IF = 9

FIRST OPTIMIZATION ANALYSIS BY COMPUTER WITH
THE 9 INITIAL TEST CONDITIONS:

THE PERFORMANCE INDEX IS DEFINED AS:

$$PI=Q*PR+(1.-Q)*(1./CU) \quad WHERE \quad Q=0.0$$

	SPINDLE SPEED (N) (RPM)	FEED (F) (IPR)	COST INDEX (CU)	PRODUCTION RATE INDEX (PR)
(1)	160.0	0.00680	2.200	0.05
(2)	220.0	0.00680	1.693	0.06
(3)	260.0	0.00680	1.511	0.07
(4)	160.0	0.01020	1.525	0.07
(5)	160.0	0.00920	1.627	0.06
(6)	220.0	0.01020	1.182	0.09
(7)	220.0	0.00920	1.290	0.08
(8)	260.0	0.01020	1.059	0.11
(9)	260.0	0.00920	1.175	0.10

BASED UPON THE ABOVE PRODUCTION TEST RESULTS, THE FOLLOWING CUTTING
CONDITION IS RECOMMENDED AS THE OPTIMUM CONDITION TO BE TESTED FURTHER:

```
***********************************
*                                 *
* SPINDLE SPEED (NOP): 320.0 RPM  *
*                                 *
* FEED (FOP):          0.0094 IPR *
*                                 *
* CUTTING SPEED (VOP): 502.7 FPM  *
*                                 *
***********************************
```

FURTHER TESTING CONDITIONS RECOMMENDED BASED UPON THE OPTIMUM
CONDITION FROM THE PREVIOUS TEST RESULTS:

	SPINDLE SPEED (N) (RPM)	FEED (F) (IPR)
(1)	320.0	0.0094
(2)	280.0	0.0094
(3)	240.0	0.0094

PROCEDURE:

    (A) TEST ALL 3 CONDITIONS OF THE ABOVE RECOMMENDED COMBINATION
    OF SPEEDS AND FEEDS AT THE WORKSHOP UNDER THE PRODUCTION
    CONDITIONS

    (B) COLLECT AND RECORD THE FOLLOWING PRODUCTION DATA OBTAINED
    FROM THE SHOP AFTER THE PRODUCTION TESTS AS SPECIFIED:

        (1) NUMBER OF PARTS PRODUCED (NP)
        DURING A SPECIFIED TIME PERIOD (T)

        (2) NUMBER OF TOOL CHANGES (NTC) MADE
        DURING A SPECIFIED TIME PERIOD (T)

    (C) FEED BACK THE ABOVE INFORMATION BY PUNCHING INPUT CARDS
    AS SPECIFIED IN THE WRITE-UP.

CODES FOR NEXT RUN OF THE COMPUTER PROGRAM:

    IR = 1
    IF = 3

(CONTINUED)

FURTHER OPTIMIZATION ANALYSIS BY COMPUTER WITH THE PRODUCTION TEST RESULTS UNDER THE PREVIOUSLY RECOMMENDED TEST CONDITIONS:

ANALYSIS # 1

THE PERFORMANCE INDEX IS DEFINED AS:

$$PI=Q*PR+(1.-Q)*(1./CU) \quad WHERE \quad Q=0.0$$

	SPINDLE SPEED (N) (RPM)	FEED (F) (IPR)	COST INDEX (CU)	PRODUCTION RATE INDEX (PR)
(1)	320.0	0.00940	1.193	0.13
(2)	280.0	0.00940	1.162	0.11
(3)	240.0	0.00940	1.165	0.10

BASED UPON THE ABOVE PRODUCTION TEST RESULTS, THE FOLLOWING CUTTING CONDITION IS RECOMMENDED AS THE OPTIMUM CONDITION TO BE TESTED FURTHER:

```
************************************
*                                  *
* SPINDLE SPEED (NOP):  300.0 RPM  *
*                                  *
* FEED (FOP):           0.0102 IPR *
*                                  *
* CUTTING SPEED (VOP):  471.2 FPM  *
*                                  *
************************************
```

FURTHER TESTING CONDITIONS RECOMMENDED BASED UPON THE OPTIMUM CONDITION FROM THE PREVIOUS TEST RESULTS:

	SPINDLE SPEED (N) (RPM)	FEED (F) (IPR)
(1)	300.0	0.0102
(2)	260.0	0.0102
(3)	320.0	0.0094
(4)	220.0	0.0102

PROCEDURE:

    (A) TEST ALL 4 CONDITIONS OF THE ABOVE RECOMMENDED COMBINATION OF SPEEDS AND FEEDS AT THE WORKSHOP UNDER THE PRODUCTION CONDITIONS.

    (B) COLLECT AND RECORD THE FOLLOWING PRODUCTION DATA OBTAINED FROM THE SHOP AFTER THE PRODUCTION TESTS AS SPECIFIED:

        (1) NUMBER OF PARTS PRODUCED (NP) DURING A SPECIFIED TIME PERIOD (T)

        (2) NUMBER OF TOOL CHANGES (NTC) MADE DURING A SPECIFIED TIME PERIOD (T)

    (C) FEED BACK THE ABOVE INFORMATION BY PUNCHING INPUT CARDS AS SPECIFIED IN THE WRITE-UP.

CODES FOR NEXT RUN OF THE COMPUTER PROGRAM:

    IR = 1
    IF = 4

FURTHER OPTIMIZATION ANALYSIS BY COMPUTER WITH THE PRODUCTION
TEST RESULTS UNDER THE PREVIOUSLY RECOMMENDED TEST CONDITIONS:

ANALYSIS # 2

THE PERFORMANCE INDEX IS DEFINED AS:

$$PI=Q*PR+(1.-Q)*(1./CU) \quad WHERE \quad Q=0.0$$

	SPINDLE SPEED (N) (RPM)	FEED (F) (IPR)	COST INDEX (CU)	PRODUCTION RATE INDEX (PR)
(1)	300.0	0.01020	1.077	0.13
(2)	260.0	0.01020	1.059	0.11
(3)	320.0	0.00940	1.193	0.13
(4)	220.0	0.01020	1.182	0.09

BASED UPON THE ABOVE PRODUCTION TEST RESULTS, THE FOLLOWING CUTTING
CONDITION IS RECOMMENDED AS THE OPTIMUM CONDITION TO BE TESTED FURTHER:

```
*********************************
*                               *
* SPINDLE SPEED (NOP):  300.0 RPM *
* FEED (FOP):          0.0102 IPR *
*                               *
* CUTTING SPEED (VOP):  471.2 FPM *
*********************************
```

```
************************************************************
*                                                          *
* THE OPTIMUM POINT ABOVE HAS SATISFIED THE STOPPING       *
* CRITERION.  NO FURTHER TESTING IS NEEDED.  THE           *
* ABOVE MACHINING PARAMETERS CAN BE CONSIDERED AS THE      *
* OPTIMUM CONDITION FOR THE GIVEN MACHINING OPERATION.     *
************************************************************
```
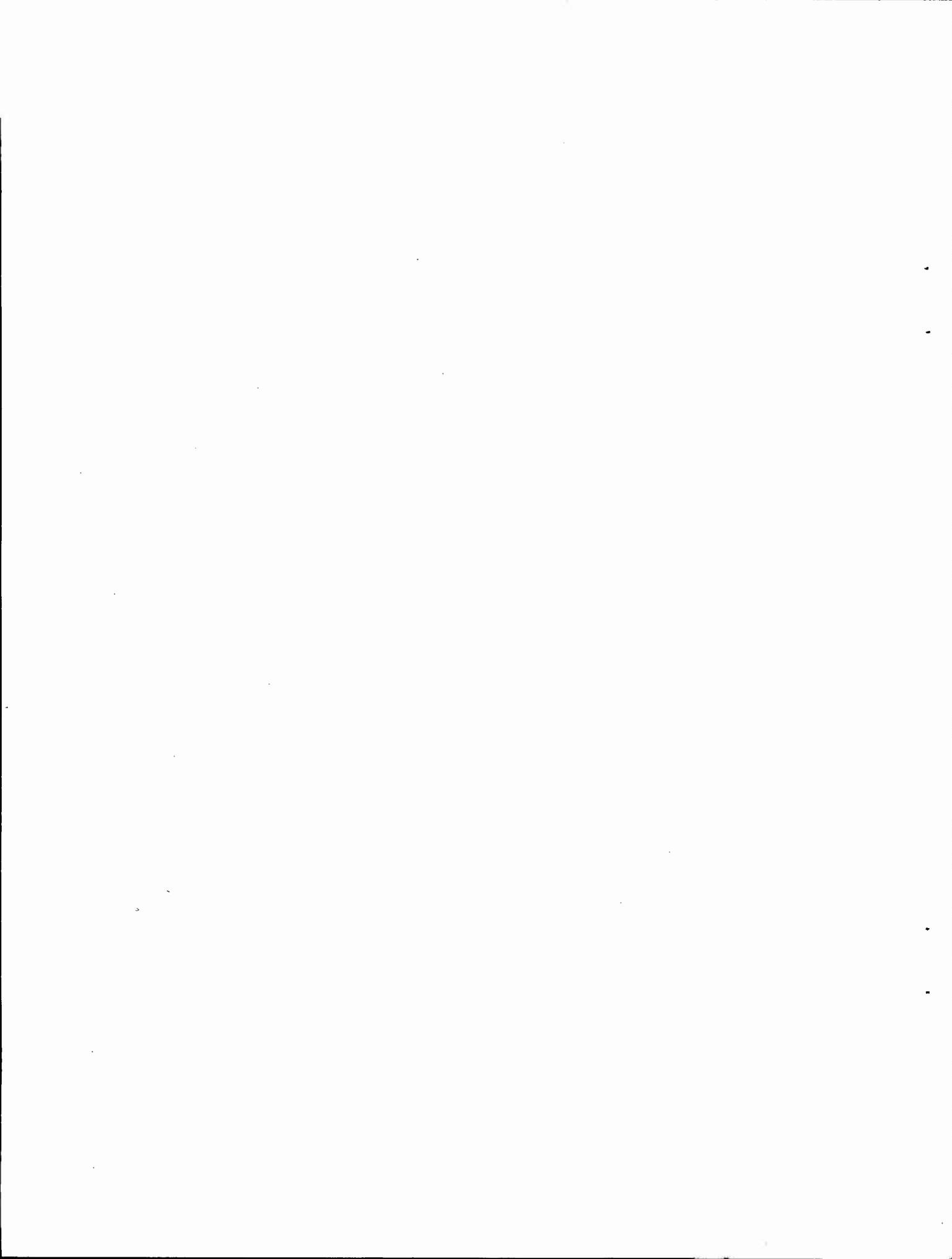
APPENDIX B. PRODUCTION OPTIMIZATION METHOD

FLOW DIAGRAM, POM COMPUTER PROGRAM

75
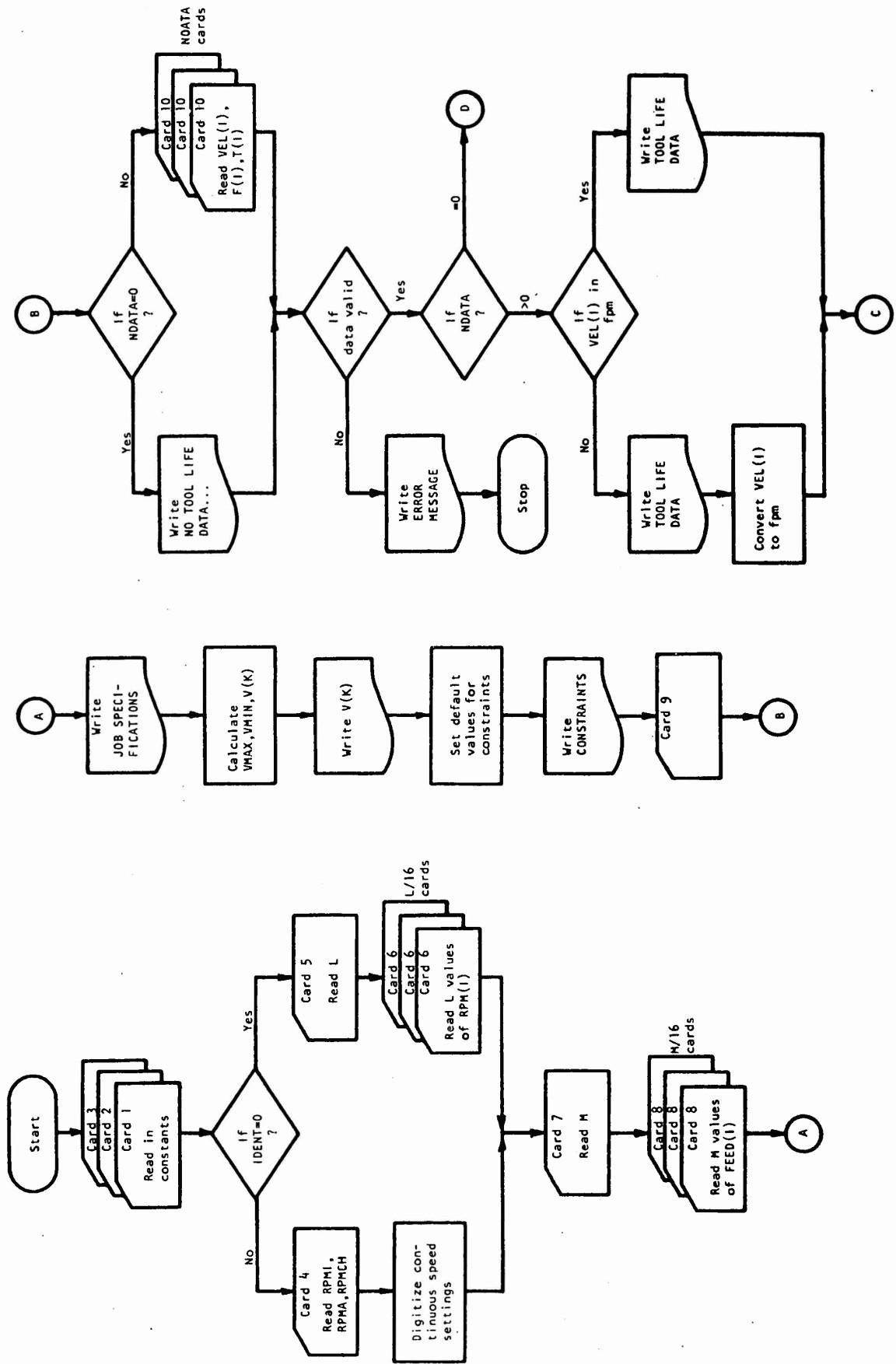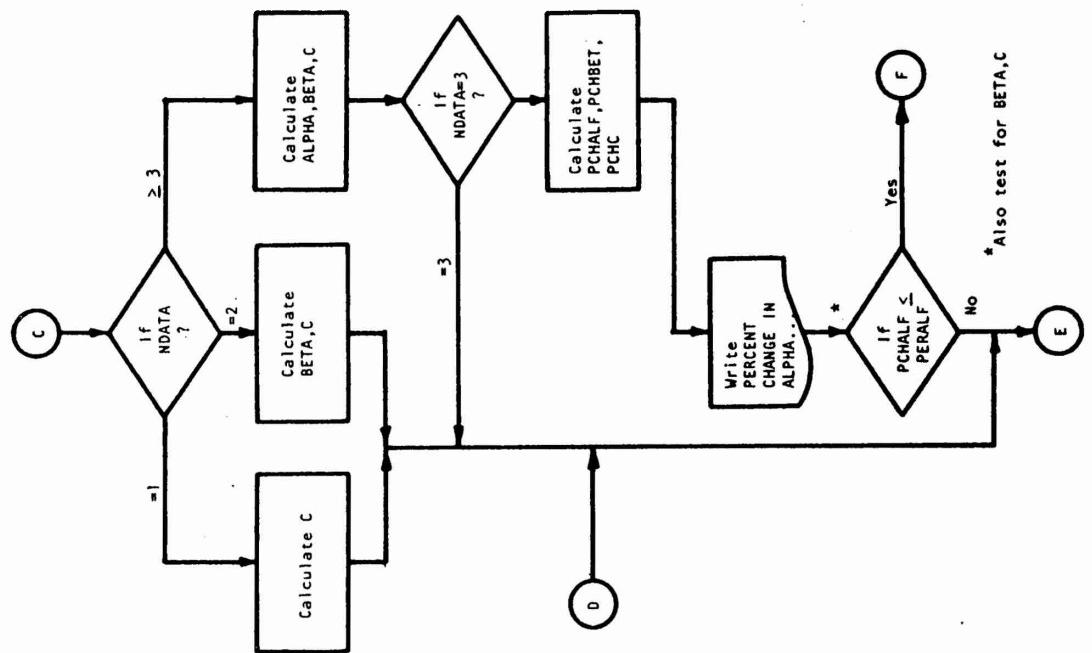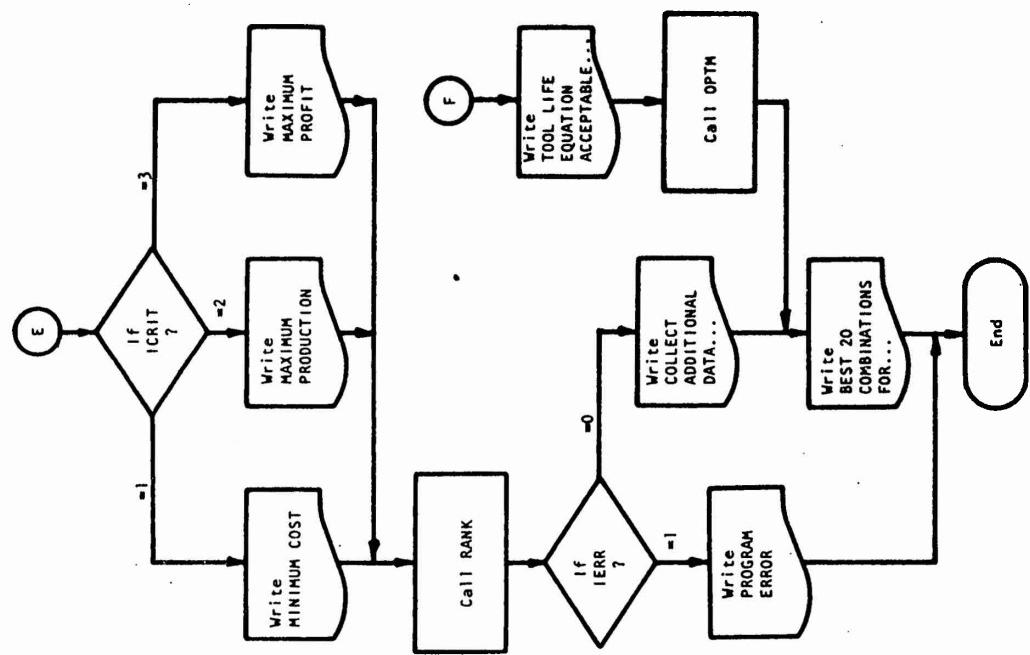
FORMAT GUIDE FOR POM INPUT DATA CARDS

Card Type	Variable	Column Number	Number Format	Card Type	Variable	Column Number	Number Format
1	NPR	1-5	I5	6	L values,	1-5	F5.0
	IDENT	6-10	I5		RPM(I)	to	
	DCMIN	11-20	F10.0			76-80	F5.0
	DCMAX	21-30	F10.0		Must be ordered from low to high.		
	VLOW	31-40	F10.0				
	VHIGH	41-50	F10.0	7	M	1-3	I3
	FLOW	51-60	F10.0				
	FHIGH	61-70	F10.0	8	M values,	1-5	F5.0
					FEED(I)	to	
2	XL	1-10	F10.0			76-80	F5.0
	DIAS	11-20	F10.0		Must be ordered from low to high.		
	DIAF	21-30	F10.0		If M is greater than 16, continue		
	FR	31-40	F10.0		on another card, same columns.		
	T1	41-50	F10.0				
	TL	51-60	F10.0	9	NDATA	1-10	I10
	TCT	61-70	F10.0		ICRIT	11-20	I10
	TGC	71-80	F10.0		NVTYPE	21-30	I10
					ALPHA	31-40	F10.0
3	PRICE	1-10	F10.0		BETA	41-50	F10.0
	COSTMT	11-20	F10.0		C	51-60	F10.0
	FEEDRF	21-30	F10.0		PERALF	61-65	F5.0
	CO	31-40	F10.0		PERBET	66-70	F5.0
	CE	41-50	F10.0		PERC	71-75	F5.0
	SHP	51-60	F10.0				
	HPM	61-70	F10.0	10	VEL(I)	1-10	F10.0
	EFF	71-80	F10.0		F(I)	11-20	F10.0
					T(I)	21-30	F10.0
4	RPMI	1-10	F10.0		Include only when NDATA ≠ 0.		
	RPMA	11-20	F10.0		One card for each set of data.		
	RPMCH	21-30	F10.0				
	Include only if IDENT = 1.						
5	L	1-3	I3				
	Include only if IDENT = 0.						

Format I5: Integer-valued number, no more than 5 digits or 4 digits and minus sign, no decimal point. Right justified. Read blank as 0. Examples:

0	1	2	0	0	read	1200
-	1	2	0	0	read	-1200
		2	0	0	read	200
		2			read	200
				2	read	2
					read	0

Format F10.0: Real-valued number, no more than 9 digits and decimal point. Minus sign may precede. Computer prints decimal point in 10th column if number has none. Read blank as 0. Examples:

		1	0	.						read	10.
						1	0	.		read	10.
1	3	.	7	3	3					read	13.733
				1	0	0	.			read	100.
										read	0

POM COMPUTER PROGRAM LISTING

PRODUCTION OPTIMIZATION METHOD

```
C     THE PURPOSE OF THIS PROGRAM IS TO DETERMINE
C     THE TAYLOR'S TOOL LIFE EQUATION FOR A PARTICU-
C     LAR OPERATION USING ACTUAL PRODUCTION SHOP DA-
C     TA. AFTER INITIAL ESTIMATES HAVE BEEN MADE FOR
C     THE VALUES OF THE TOOL LIFE PARAMETERS, ALPHA,
C     BETA, AND C, A TEST RECOMMENDATION IS MADE.
C     THE TEST RESULTS ARE THEN RETURNED TO THE PRO-
C     GRAM. THE ESTIMATES ARE REFINED AND A NEW
C     TEST RECOMMENDATION IS MADE. THIS PROCESS IS
C     CONTINUED UNTIL THE TOOL LIFE PARAMETERS HAVE
C     BEEN DETERMINED. A COMPLETE OPTIMIZATION ANAL-
C     YSIS IS THEN MADE WITH THE FINAL RESULTS.

      IMPLICIT REAL*8 (A-H,O-Z)

      DIMENSION RPM(100),FEED(100),VII(100),XI3(1),QX(3,1)
     1F(100),QX(3,1)
      COMMON/A1/L,M,IDENT/A2/DIAS,DIAF,DCMAX,DCMIN,FLOW,FHIGH,FFEDRF,VLD
     1W,VHIGH,EFF,SHP,HPM,XL,C,ALPHA,BETA,T1,FR,TL,TGC,CO,TCT,CE,PRICE,C
     2OSTMT/A3/RPM,FEED,V
    5 RFAO(5,10,END=1000)NPR,IDENT ,DCMIN,DCMAX,VLOW,VHIGH,FLOW,FHIGH,XL
     1,DIAS,DIAF,FR,T1,TL,TCT,TGC
   10 FORMAT(2I5,6F10.0/8F10.0)
      READ 15,PRICE,COSTMT,FFEDRF,CO,CE,SHP,HPM,EFF
   15 FORMAT(8F10.0)
      IF(IDENT.EQ.0)GOTO 30
      READ 15,RPM1,RPM4,RPMCH
      XSTEPS=IRPM4-RPM1)/RPMCH+1.D0
      L=XSTEPS
      IF(XSTEPS-L.GT.0.00)L=L+1
      DO 25 I=1,L
      XS=I
   25 RPM(I)=RPM1+(XS-1.D0)*RPMCH
      GO TO 35
   30 READ 40,L,(RPM(I),I=1,L)
   35 READ 40,M,(FEED(I),I=1,M)
   40 FORMAT(I3/16F5.0)
      PRINT 50,NPR,XL,DIAS,DIAF,FR,T1,TL,TCT,TGC,CO,CE,COSTMT,PRICE,
     1SHP,HPM,EFF,C,IRPM(I),I=1,L)
   50 FORMAT('(1)PROJECT NO.'I5/1X,'JOB SPECIFICATIONS'/6X,'LENGTH OF PAR
     1T TO BE MACHINED'4X,'L ='F8.3,'IN.'/6X,'DIAMETER TO START WITH'8X,
     1'DIAS ='F8.3,'IN.'/6X,'DIAMETER TO FINISH AT'9X,'DIAF ='F8.3,'IN.'
     1/6X,'RETURN SPEED OF THE CARRIAGE'4X,'FR ='F8.0,'IN/MIN.'/6X,'TIM
     1E FOR FORWARD OR BACKWARD MOTION AT'/8X,'THE START OR END OF A CUT
     1'5X,'T1 ='F8.5,'MIN.'/6X,'TIME FOR LOADING AND/12X,'UNLOADING THE
     1 WORKPIECE  TL ='F8.4,'MIN.'/6X,'TOOL CHANGING TIME'13X,'TCT ='F8
     1.4,'MIN.'/6X,'TIME FOR CHANGING GEARS'8X,'TGC ='F8.4,'MIN.'/1X,'C
     1ST FACTORS'/6X,'LABOR RATE PLUS OVERHEAD'8X,'CO = ',F8.5,'  $/MI
     1N.'/6X,'TOOL COST PER EDGE',14X,'CE = ',F8.5,'  $/EDGE'/,6X,'MATE
     1RIAL COST',15X,'COSTMT = ',F9.4,'  $/PIECE'/,6X,'SELLING PRICE',16
     1X, 'PRICE = ',F9.4,' $/PIECE'/,
     1PECIFIC HORSE POWER'11X,'SHP =F8.3,'H.P./CU.IN/MIN.'/1X,'MACHINE
     1 TOOL TO BE USED'/6X,'AVAILABLE HORSE POWER'10X,'HPM ='F8.3,'H.P.'
```

```
     1/6X,'EFFICIENCY OF DRIVE ENERGY'5X,'EFF ='F8.5/6X,'SPINDLE SPEEDS
     1AVAILABLE IN R.P.M.'1N'14,' STEPS'/110X,23F6.3))
      PRINT 55,M,(FEED(J),J=1,M)
   55 FORMAT(6X,'FEEDS AVAILABLE IN IN./REV.'1N'14,' STEPS'/110X,12F10.
     14))
      VMAX=3.14159265D0*DIAF*RPM(L)/12.D0
      VMIN=3.14159265D0*DIAS*RPM(1)/12.D0
      DO 60 K=1,L
   60 V(K)=VMIN*DFXP((K-1.D0)/(L-1.D0)*DLOG(VMAX/VMIN))
      PRINT 65,(V(K),K=1,L)
   65 FORMAT ('0SPEEDS TESTED IN F.P.M.'/110X,20F6.0))
C*****WHEN THE UPPER LIMITS FOR SPEED AND/OR FEED ARE NOT GIVEN, THE
C*****INSTRUCTIONS THAT FOLLOW WILL SET THESE LIMITS.
      IF(VHIGH.GT.0.000)GO TO 75
      VHIGH=V(1)
   70 L=2,L
      IF(V(I).GT.VHIGH)VHIGH=V(I)
   70 CONTINUE
   75 IF(FHIGH.GT.0.D00) GO TO 85
      FHIGH=FEED(1)
      DO 80 I=2,M
      IF(FEED(I).GT.FHIGH)FHIGH=FEED(I)
   80 CONTINUE
   85 PRINT 90,DCMIN,DCMAX,VLOW,VHIGH,FLOW,FHIGH
   90 FORMAT(//,' CONSTRAINTS ON CUTTING PARAMETERS:',/,6X,'DEPTH OF CU
     1T(INCHES): ',F8.5,' < D < ',F8.5,/,6X,'CUTTING SPEED (FPM): ',F6.
     11,' < V < ',F6.1,/,6X,'FEED (IN./REV.): ',F7.5,' < F < ',F7.5,//)
      READ 86, NDATA,ICRIT,NVTYPE, ALPHA,BETA,C,PERALF,PERRET,PERC
   86 FORMAT(3I10,3F10.0,3F5.0)
      ALPHA1=ALPHA
      BETA1=BETA
      C1=C
      IF(NDATA-1)87,88,88
   87 PRINT 97, ALPHA,BETA,C
   97 FORMAT('NO TOOL LIFE DATA IS AVAILABLE BUT THE PARAMETERS OF THE
     1TOOL LIFE EQUATION ARE ASSUMED TO BE:',/,T6,'TOOL LIFE EXPONENT: A
     2LPHA = ',F7.4,/,T11,'FEED EXPONENT:   BETA = ',F7.4,/,T6,'TOOL LIFE
     3 CONSTANT:       C = ',F8.3)
      GO TO 141
   88 READ 96, (VEL(I),FII(I),TII(I),I=1,NDATA)
   96 FORMAT(3F10.0)
  141 IF (IXL.LE.0.D0.OR.DIAS-DIAF.LE.0.D0.OR.DIAS.LE.0.D0.OR.DIAS.LE.0.D0) GO TO 470
      IF (TCT.LE.0.D0.OR.TGC.LE.0.D0.OR.DO.OR.CO.LE.0.D0.OR.CE.LE.0.D0) GO TO
     1470
      IF (C.LE.0.D0.OR.SHP.LE.0.D0.OR.HPM.LE.0.D0.OR.EFF.LE.0.D0) GO TO
     1470
      IF (EFF.GT.1.D0.OR.L.LE.0.OR.M.LE.0.OR.VLOW.GE.VHIGH) GO TO 470
      IF (FLOW.GE.FHIGH.OR.RPM1.GE.RPM4.OR.RPMCH.LE.0.D0) GO TO 470
      IF (RPM1.LT.0.D0.OR.RPM4.LE.0.D0) GO TO 470
      IF(NDATA.EQ.0)GOTO 971
      IF(NVTYPE.EQ.1) GO TO 66
      PRINT 77, (I,VEL(I),FII(I),TII(I),I=1,NDATA)
   46 VEL(I)=VEL(I)*DIAS*3.14159265D0/12.D0
      DO 46 I=1,NDATA
      GOTO 98
   66 PRINT 76, (I,VEL(I),FII(I),TII(I),I=1,NDATA)
```

```fortran
142 PCHALF=DABS((ALPHA1-ALPHA)/ALPHA1)*100.D0
    PCHBET=DABS((BETA1-BETA)/BETA1)*100.D0
    PCHC=DABS((C1-C)/C1)*100.D0
    PRINT 1042, PCHALF,PERALF,PCHBET,PERBET,PCHC,PERC
1042 FORMAT('-WITH THE NEW TOOL LIFE EQUATION, THE PERCENTAGE CHANGES I
   1N THE PARAMETERS OF THE TOOL LIFE EQUATION ARE:',/,T6,'CHANGE IN "
   2ALPHA" = ',F8.3,'% (MAXIMUM VARIATION ALLOWED IS ',F5.2,'%)',/,T7,
   3'CHANGE IN "BETA" = ',F8.3,'% (MAXIMUM VARIATION ALLOWED IS ',F5.2
   4,'%)',/,T10,'CHANGE IN "C" = ',F8.3,'% (MAXIMUM VARIATION ALLOWED
   51S',F5.2,'%)')
    IF(PCHALF.LE.PERALF.AND.PCHBET.LE.PERBET.AND.PCHC.LE.PERC)GOTO 143
971 IF(ICRIT-2)151,152,153
151 PRINT 161
161 FORMAT('-ON THE ASSUMPTION THAT THIS TOOL LIFE EQUATION IS CORRECT
   1, THE FOLLOWING TABLES GIVE THE BEST CONDITIONS FOR "MINIMUM COST"
   2')
    CALL RANK(1,IERR)
    GO TO 171
152 PRINT 162
162 FORMAT('-ON THE ASSUMPTION THAT THIS TOOL LIFE EQUATION IS CORRECT
   1, THE FOLLOWING TABLES GIVE THE BEST CONDITIONS FOR "MAXIMUM PRODU
   2CTION RATE"')
    CALL RANK(2,IERR)
    GO TO 171
153 PRINT 163
163 FORMAT('-ON THE ASSUMPTION THAT THIS TOOL LIFE EQUATION IS CORRECT
   1, THE FOLLOWING TABLES GIVE THE BEST CONDITIONS FOR "MAXIMUM PROFI
   2T RATE"')
    CALL RANK(3,IERR)
171 IF(IERR.EQ.1) GOTO 5
630 PRINT 640
640 FORMAT(///,' *****************************************************
   1************',//,T27,'IT IS NECESSARY TO COLLECT ADDITIONAL TOOL L
   1IFE INFORMATION ON THIS OPERATION',//,T32,'BEFORE RELIABLE TOOL LIF
   1E DATA CAN BE OBTAINED FOR FURTHER OPTIMIZATION',//,' **********
   1******************',//,T10,'TRY
   1ONE OF THE RECOMMENDED CUTTING CONDITIONS AND SUPPLY THE RESULTING
   1 TOOL LIFE DATA TO THIS PROGRAM FOR PROCESSING',//,T17,'- A NEW TOO
   1L LIFE EQUATION WILL BE CALCULATED AND A NEW SET OF CUTTING CONDIT
   1IONS WILL BE GIVEN -',
   1    //,' ****************************************************
   1**********')
    GO TO 5
143 PRINT 181
181 FORMAT('-*****************************************************
   2*******',//,T21,'THE PERCENTAGE CHANGES ON THE PARAMETERS OF THE T
   3OOL LIFE EQUATION ARE BELOW THE SPECIFIED',/,T21,'MAXIMUM LIMITS.
   4THEREFORE THIS TOOL LIFE EQUATION CAN BE ACCEPTED AS CORRECT.',/,
   5T11,'THE FOLLOWING TABLES GIVE THE BEST CONDITIONS FOR MINIMUM COS
   6T, MAXIMUM PRODUCTION RATE, AND MAXIMUM PROFIT RATE.',//,' *******
   7*****************************************************')
    CALL OPTM
    GO TO 5
470 PRINT 471
471 FORMAT('-********************PARAMETER ERROR*****JOB ABANDONED*******')
    GO TO 5
```

```fortran
76 FORMAT(T26,'TOOL LIFE DATA',/,T26,'--------------',//,T27,'CUTTING
   1',/,T12,'OBSERVATION    SPEED    FEED      TOOL LIFE',/,T14,'NUMB
   2ER',8X,'(FPM)    (IPR)    (MINUTES)',/,(T15,I3,9X,F7.2,3X,F6.4,4X
   3,F9.4))
77 FORMAT(T26,'TOOL LIFE DATA',/,T26,'--------------',//,T27,'CUTTING
   1',/,T12,'OBSERVATION    SPEED    FEED      TOOL LIFE',/,T14,'NUMB
   2ER',8X,'(PPM)    (IPR)    (MINUTES)',/,(T15,I3,9X,F7.2,3X,F6.4,4X
   3,F9.4))
98 IF(NDATA-2)101,102,103
101 C=VEL(1)*T(1)**ALPHA*F(1)**BETA
    PRINT 106, ALPHA,BETA,C
106 FORMAT('-WITH ONLY ONE SET OF TOOL LIFE DATA AVAILABLE AND THE PAR
   1AMETERS  ALPHA  AND  BETA  OF THE TOOL LIFE EQUATION ASSUMED TO BE
   2:',/,T6,'TOOL LIFE EXPONENT: ALPHA = ',F7.4,/,T11,'FEED EXPONENT:
   3 BETA = ',F7.4,/,T3,'THE TOOL LIFE CONSTANT, C, IS ESTIMATED TO BE
   4:',/,T30,'C = ',F8.3)
    GOTO 971
102 BETA=(DLOG(VEL(2)/VEL(1))+ALPHA*DLOG(T(2)/T(1)))/DLOG(F(1)/F(2))
    C=VEL(1)*T(1)**ALPHA*F(1)**BETA
    PRINT 107, ALPHA,BETA,C
107 FORMAT('-WITH ONLY TWO SETS OF TOOL LIFE DATA AVAILABLE AND THE PA
   1RAMETER  ALPHA  OF THE TOOL LIFE EQUATION ASSUMED TO BE:',/,T6,'TO
   2OL LIFE EXPONENT: ALPHA = ',F7.4,/,T3,'THE PARAMETERS  BETA  AND
   3C  ARE ESTIMATED TO BE:',/,T11,'FEED EXPONENT:   BETA = ',F7.4,/,T6
   4,'TOOL LIFE CONSTANT:       C = ',F8.3)
    GO TO 971
103 DO 410 KK=1,3
    X(KK,1)=0.D0
    DO 410 KJ=1,3
410 Q(KK,KJ)=0.D0
    Q(1,1)=NDATA
    DO 420 KDATUM=1,NDATA
    QT=DLOG(T(KDATUM))
    QV=DLOG(VEL(KDATUM))
    QF=DLOG(F(KDATUM))
    Q(1,2)=Q(1,2)-QT
    Q(1,3)=Q(1,3)-QF
    Q(2,1)=Q(2,1)+QF
    Q(2,2)=Q(2,2)-QF**2
    Q(2,3)=Q(2,3)-QT**2
    Q(3,2)=Q(3,2)-QT*QF
    Q(3,3)=Q(3,3)-QF**2
    X(1,1)=X(1,1)+QV
    X(3,1)=X(3,1)+QV*QT
420 X(2,1)=X(2,1)+QV*QF
    Q(3,1)=-Q(1,2)
    CALL DMXINV(Q,3,3)
    DO 6622 KX=1,3
6622 QX(KX,1)=0.D0
    DO 6633 KX=1,3
    DO 6633 KY=1,3
6633 QX(KX,1)=QX(KX,1)+Q(KX,KY)*X(KY,1)
    C=DEXP(QX(1,1))
    ALPHA=QX(2,1)
    BETA=QX(3,1)
    PRINT 108, NDATA,ALPHA,BETA,C
108 FORMAT('-BASED ON THE ',I3,' OBSERVATIONS AVAILABLE, THE PARAMETER
   1S OF THE TOOL LIFE EQUATION ARE:',/,T6,'TOOL LIFE EXPONENT: ALPHA
   2= ',F7.4,/,T11,'FEED EXPONENT:   BETA = ',F7.4,/,T6,'TOOL LIFE CONS
   3TANT:       C = ',F8.3)
    IF(NDATA-3)971,971,142
```

```
1000 CONTINUE
     STOP
     END
     SUBROUTINE RANK(ICRIT,IERR)
     IMPLICIT REAL*8 (A-H,O-Z)
     DIMENSION IIR(2),RPM(100),FEED(100),VT(100), A(21,12,2), IA(21,2)
     COMMON/A1/L,M,IDENT/A2/DIAS,DIAF,DCMAX,DCMIN,FLOW,FHIGH,FEEDRF,VL,
    1M,VHIGH,EFF,SHP,HPM,XL,C,ALPHA,BETA,TI,FR,TL,TGC,CO,TCT,CE,PRICE,C
20   STMT/A3/RPM,FEED,V
     NMIN=(DIAS-DIAF)/DCMAX/2.D0
     IF (NMIN.LT.1)NMIN=1
     N=(DIAS-DIAF)/DCMIN/2.D0
     IF (N-25) 110,110,100
100  N=25
110  IIR(1)=0
     IIR(2)=0
     DO 405 K=NMIN,N
     DP=(DIAS-DIAF)/2.D0/K
     DO 400 J=1,M
     IF (FEED(J).LT.FLOW.OR.FEED(J).GT.FHIGH) GO TO 400
     IF (FEED(J).GE.FEEDRF) GO TO 125
     IDCON=1
     GO TO 125
120  IDCON=2
125  II=IIR(IDCON)
     DO 395 L=1,L
     IF (V(I).LT.VLOW.OR.V(I).GT.VHIGH) GO TO 395
     HP=12.D0*V(I)*FEED(J)*DP*SHP/EFF
     IF (HP-HPM) 140,140,395
140  TCUT=0.D0
     NGC=0
     VT=V(I)
     IF (IDENT) 460,145,160
145  RPMS=RPMN(RPM,VT,DIAS,L)
     RPMA=RPMS
     DO 155 KK=1,K
     DK=DIAS-2.D0*(KK-1)*DP
     RPMT=RPMN(RPM,VT,DK,L)
     TCUT=TCUT+XL/RPMT/FEED(J)
     IF (RPMB-RPMT) 150,155,150
150  RPMB=RPMT
     NGC=NGC+1
155  CONTINUE
     GO TO 170
160  RPMS=VT/3.14159265D0/DIAS*12.D0
     DO 165 KK=1,K
     DK=DIAS-2.D0*(KK-1)*DP
     RPMT=VT/3.14159265D0/DK*12.00
     TCUT=TCUT+XL/RPMT/FEED(J)
     NGC=NGC+1
165  NGC=NGC+1
170  IF (NGC) 460,180,175
175  NGC=NGC+1
180  TLIFE=(C/(V(I)*FEED(J)**BETA))**(1.D0/ALPHA)
     XNPPT=TLIFE/TCUT
     COST=CO*(K*(2.D0*TI+XL/FR)+TL+NGC*TGC)+CO*TCUT+(CO*TCT +CE)/XNPPT
     RMRT=12.D0*V(I)*FEED(J)*OP
     TIME=K*(2.D0*TI+XL/FR)+TL+NGC*TGC*TCUT+TCT/XNPPT
     PRODRT=60.D0/TIME
     PROFIT=(PRICE-COSTMT-COST)/TIME
190  IF (II) 460,190,194
```

```
     GO TO 220
194  IF(ICPIT-2)195,196,197
195  DO 215 JJ=1,II
200  IF (A(JJ,10,IDCON)-COST)  215,200,200
     M1=II+JJ+1-MM
210  DO 205 L1=1,12
205  A(M1,L1,IDCON)=A(M1-1,L1,IDCON)
210  IA(M1,IDCON)=IA(M1-1,IDCON)
     M1=JJ
     GO TO 220
215  CONTINUE
     M1=II+1
     GO TO 220
196  DO 216 JJ=1,II
     IF(A(JJ,9,IDCON)-TIME)216,201,201
201  DO 211 MM=JJ,II
     M1=II+JJ+1-MM
     DO 206 L1=1,12
206  A(M1,L1,IDCON)=A(M1-1,L1,IDCON)
211  IA(M1,IDCON)=IA(M1-1,IDCON)
     M1=JJ
     GOTO 220
216  CONTINUE
     M1=II+1
     GOTO 220
197  DO 217 JJ=1,II
     IF(PROFIT-A(JJ,12,IDCON))217,202,202
202  DO 212 MM=JJ,II
     M1=II+JJ+1-MM
     DO 207 L1=1,12
207  A(M1,L1,IDCON)=A(M1-1,L1,IDCON)
212  IA(M1,IDCON)=IA(M-1,IDCON)
     M1=JJ
     GOTO 220
217  CONTINUE
     M1=II+1
220  A(M1,1,IDCON)=V(I)
     A(M1,2,IDCON)=FEED(J)
     A(M1,3,IDCON)=OP
     IA(M1,IDCON)=K
     A(M1,4,IDCON)=RPMS
     A(M1,5,IDCON)=RPMT
     A(M1,6,IDCON)=TLIFE
     A(M1,7,IDCON)=RMRT
     A(M1,8,IDCON)=HP
     A(M1,9,IDCON)=TIME
     A(M1,10,IDCON)=COST
     A(M1,11,IDCON)=PRODRT
     A(M1,12,IDCON)=PROFIT
     IF (II-20)  295,395,460
295  II=II+1
395  CONTINUE
     IIR(IDCON)=II
400  CONTINUE
405  CONTINUE
     DO 304 IDCON=1,2
     II=IIR(IDCON)
     IF(II.EQ.0) GO TO 304
     GO TO (410,420),IDCON
410  PRINT 415,FEEDRF
```

79

```
415   FDRMAT ('1',T50,'******************************',/,T50,'*      (FEED UNDER ',F6.4,' IPR) *',/,T
      1HING CONDITIONS   *',/,T50,'*
      150,'******************************',/,///)
420   PRINT 425,FEEDRF
425   FDRMAT ('1',T51,'*******************************',/,T51,'** II.  ROUGHIN
      1G CONDITIONS *',/,T51,'*     (FEED EQUAL TO DR  *',/,T51,'*       A
      1BOVE ',F6.4,' IPR)   *',/,T51,'******************************',/,///)
430   IF(ICRIT-2)301,302,303
301   PRINT 435,II
      GOTO 3D4
302   PRINT 445,II
      GDTD 3D4
303   PRINT 450
435   FDRMAT (' MINIMUM COST RANKING',10X,'BFST ',I3,' COMBINATIONS',/,T
      1106,'*******',/,T115,'* *')
445   FDRMAT (' MAXIMUM PRODUCTION RANKING',10X,'BEST ',I3,' CDMBINATION
      1S',/,T115,'*******',/,T115,'* *')
450   FDRMAT (' MAXIMUM PRDFIT RANKING',10X,'BEST ',I3,' CDMBINATIONS',T
      1124,'*******',/,TI24,'* *')
304   PRINT 440,(I,A(I,1,IDCON),A(I,2,IDCON),A(I,3,IDCON),IA(I,IDCON),A(
      1I,4,IDCON),A(I,5,IDCON),A(I,6,IDCON),A(I,7,)DCON),A(I,B,IDCON),A(I
      1,9,IDCON),A(I,10,IDCON),A(I,11,IDCON),A(I,12,IDCON),I=1,II)
440   FDRMAT (6X,'CUTTING',13X,'DEPTH   ND.    RPM TD   RPM TD',15X,'RE4
      1DVAL     HORSE',8X,'TIME      COST   PROD.  PRDFIT',/,' RANK
      1SPEED      FEED',7X,'OF       START    FINISH   TOOL      RAT
      1E IN     PDWER      REQUIRED    PER      RATE    RATE',/,6X,'INTE
      1NDED',13X,'CUT    PASS    WITH      WITH       LIFE     CUTTING
      1REQU)RED   PER PIECE   PER PIECE',/,7X,'(FPM)   (IPR)       (IN)   REQ
      1.     (RPM)    (MIN)      (MIN)     (HP)      (MIN)      (MIN)',7
      1X,'($)     (PC/HR)    ($/MIN)',/,(I4,F8.0,F9.4,F10.5,I6,2F9.0,2F12.4
      1,F9.2,F13.3,F10.4,F10.3))
      IERR=C
      RETURN
460   PRINT 465,NPR
465   FDRMAT (1X,'PRDGRAM ERRDR --- PROBLEM ND. ',I3,' ABANDDNED.')
      IERR=I
      RETURN
      END
      FUNCTION RPMNIR,V,O,L)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(100)
      A=12.D0/3.141592650D*V/D
      DD 5 I=1,L
      IF (R(I).GE.A) GO TD 15
5     CONTINUE
      RPMN=R(II)
      RETURN
15    IF (R(I).EQ.A.DR.I.EQ.1) GO TD 10
      B=(R(I)+R(I-11)/2.D0
      IF (A-B)  20,20,10
1D    RPMN=R(I-1)
20    RPMN=R(I-1)
      RETURN
      END
      SUBROUTINE DPTH
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION RPM(100),FEED(100),V(100), A(21,12,2), IA(21,2), B(21,12
      1,2), IB(21,2), D(21,12,2), 10(21,2),IIR(2)
      COMMON/A1/L,M,IDENT,A2/DIAS,DIAF,DCMAX,DCMIN,FLOW,FHIGH,FEEDRF,VLD
      1M,VHIGH,EFF,SHP,HPM,XL,C,ALPHA,BETA,T1,FR,TL,TGC,CO,TCT,CE,PRICE,C
      2DSTMT/A3/RPM,FEED,V

      NMIN=(DIAS-DIAF)/DCMAX/2.D0
      IF(NMIN.LT.1)NMIN=I
      N=(DIAS-DIAF)/DCMIN/2.D0
      IF(N-25) 11D,110,100
100   N=25
110   IIR(1)=0
      IIR(2)=0
      DD 405 K=NMIN,N
      DP=(DIAS-DIAF)/2.D0/K
      DD 400 J=1,M
      IF (FEED(J).LT.FLOW.DR.FEED(J).GT.FHIGH) GD TD 400
      IF (FEED(J).GE.FEEDRF) GD TD 120
      IDCON=1
      GD TO 125
120   IDCON=2
125   II=)R(IDCON)
      IF (V(I).LT.VLDW.DR.V(I).GT.VHIGH)  GD TD 395
      HP=12.D0*V(I)*FEED(J)*DP*SHP/EFF
      IF (HP-HPM) 140,140,395
140   TCUT=0.DO
      NGC=0
      VT=V(I)
      IF (IDENT)  460,145,160
145   RPMS=RPMN(RPM,VT,DIAS,L)
      RPMB=RPMS
      DD 155 KK=1,K
      DK=DIAS-2.D0*(KK-1)*DP
      RPMT=RPMN(RPM,VT,DK,L)
      TCUT=TCUT+XL/RPMT/FEED(J)
      IF (RPMB-RPMT) 15D,155,150
150   RPMB=RPMT
155   CDNTINUE
      GD TD 170
160   RPMS=VT/3.141592650D/DIAS*12.D0
      DD 165 KK=1,K
      DK=DIAS-2.D0*(KK-1)*DP
      RPMT=VT/3.141592650D/DK*12.00
      TCUT=TCUT+XL/RPMT/FEED(J)
165   NGC=NGC+1
170   IF (NGC)    460,1B0,175
175   NGC=NGC+1
1B0   TLIFE=(C/(V(I)*FEED(J)*BETA))**(1.D0/ALPHA)
      XNPPT=TLIFE/TCUT
      COST=CD*(K*(2.D0*T1+XL/FR)+TL+NGC*TGC)+CO*TCUT+(CO*TCUT+CE)/XNPPT
      RMRT=12.D0*V(I)*XL/FR)+TL+NGC*TGC*TCUT+TCT/XNPPT
      TIME=K*(2.D0*T1+XL/FR)+TL+NGC*TGC*CO*TCUT+TCT/XNPPT
      PRODRT=60.D0/TIME
      PROFIT=(PRICE-COSTMT-COST)/TIME
      IF (II)   460,190,195
190   M1=1
      GD TD 22D
195   DD 215 JJ=1,II
      IF (A(JJ,10,IDCON)-COST)   215,200,200
200   DD 210 MM=JJ,II
      M1=I+JJ+1-MM
      DD 205 L1=1,12
205   A(M1,L1,IDCON)=A(M1-1,L1,IDCON)
210   IA(M1,IDCON)=IA(M1-1,IDCON)
      M1=JJ
```

```
      GO TO 220
215   CONTINUE
      M1=I+1
220   A(M1,1,IDCON)=V(I)
      A(M1,2,IDCON)=FEED(J)
      A(M1,3,IDCON)=OP
      IA(M1,IDCON)=K
      A(M1,4,IDCON)=RPMS
      A(M1,5,IDCON)=RPMT
      A(M1,6,IDCON)=TLIFE
      A(M1,7,IDCON)=RMRT
      A(M1,8,IDCON)=HP
      A(M1,9,IDCON)=TIME
      A(M1,10,IDCON)=COST
      A(M1,11,IDCON)=PRODRT
      A(M1,12,IDCON)=PROF(T
      IF (II) 460,225,230
225   M1=1
      GO TO 255
230   DO 250 JJ=1,II
      IF (B(JJ,9,IDCON)-T(MF) 250,235,235
235   DO 245 MM=JJ,II
      M1=I(+JJ+1-MM
      DO 240 L1=1,12
240   B(M1,L1,IDCON)=B(M1-(,L1,IDCON)
245   IB(M1,IDCON)=(B(M1-1,IDCON)
      M1=JJ
      GO TO 255
250   CONTINUE
      M1=I(+1
255   B(M1,1,IDCON)=V(I)
      B(M1,2,IDCON)=FEED(J)
      B(M1,3,IDCON)=OP
      IB(M1,IDCON)=K
      B(M1,4,IDCON)=RPMS
      B(M1,5,IDCON)=RPMT
      B(M1,6,IDCON)=TLIFE
      B(M1,7,IDCON)=RMRT
      B(M1,8,IDCON)=HP
      B(M1,9,IDCON)=TIME
      B(M1,10,IDCON)=COST
      B(M1,11,IDCON)=PRODRT
      B(M1,12,IDCON)=PROF(T
      IF (II) 460,260,265
260   M1=1
      GO TO 290
265   DO 285 JJ=1,II
      IF (PROF(T-O(JJ,12,IDCON)) 285,270,270
270   DO 280 MM=JJ,1I
      M1=I(+JJ+1-MM
      DO 275 L1=1,12
275   D(M1,L1,IDCON)=D(M1-1,L1,IDCON)
280   IO(M1,IDCON)=IO(M1-1,IDCON)
      M1=JJ
      GO TO 290
285   CONTINUE
      M1=II+1
290   O(M1,1,IDCON)=V(I)
      D(M1,2,IDCON)=FEED(J)
      D(M1,3,IDCON)=OP
      ID(M1,IDCON)=K
      O(M1,4,IDCON)=RPMS
      D(M1,5,IDCON)=RPMT
      D(M1,6,IDCON)=TLIFE
      O(M1,7,IDCON)=RMRT
      O(M1,8,IDCON)=HP
      O(M1,9,IDCON)=TIME
      O(M1,10,IDCON)=COST
      O(M1,11,IDCON)=PRODRT
      D(M1,12,IDCON)=PROF(T
      IF (II-20)  295,395,450
295   (I=(I+1
395   CONTINUE
      (IR(IDCON)=(I
400   CONTINUE
405   CONTINUE
      DO 455 IDCON=1,2
      II=IR(IDCON)
      IF(II.EQ.0) GO TO 455
      GO TO (410,420),IDCON
410   PRINT 415,FEEDRF
415   FORMAT ('1',T50,'*****************',/,T50,'*  (  FINIS
     1HING CONDITIONS    *',/,T50,'*   (FEED UNDER ',F6.4,'   (P) *',/,T
     150,'*****************',////)
      GO TO 430
420   PRINT 425,FEEDRF
425   FORMAT ('1',T51,'*****************',/,T51,'*   (I, ROUGHIN
     1G CONDITIONS *',/,T51,'*   (FEED EQUAL TO OR   *',/,T51,'*
     18DVE ',F6.4,'  (PR)   *',/,T51,'*****************',////)
430   PRINT 435,II
435   FORMAT (' MINIMUM COST RANKING',10X,'BEST ',I3,' COMBINATIONS',/,T
     1106,'*******',/,T106,'*      *')
      PRINT 440(I,A(I,1,IDCON),A(I,2,IDCON),IA(I,3,IDCON),IA(I,3,IDCON),A(
     1I,4,IDCON),A(I,5,IDCON),A(I,6,IDCON),A(I,7,IDCON),A(I,8,IDCON),A((
     1,9,IDCON),A(I,10,IDCON),A(I,11,IDCON),A(I,12,IDCON),I=1,II)
440   FORMAT (6X,'CUTTING',13X,'DEPTH   NO.   RPM TO   RPM TO',15X,'REM
     1OVAL   HORSE',8X,'TIME   COST  PROD.  PROFIT',/,' RANC
     1SPEED      FEED',7X,'OF   START   FINISH   TOOL       RAT
     1E IN    POWER  REQUIRED   PER   RATE  CUTTING
     1NOEO',13X,'CUT   PASS   WITH   LIFE     (IPR)      (IN)    REQ
     1,  .(S)    PER PIECE   PIFCE',/,7X,'(FPM)          (M(N),'7
     1X,'(S)     (PC/HR)     (MIN)    (CU IN/MIN)    (MP)     (S/MIN)',/
     1,F9.2,F13.3,F10.4,F8.0,F9.4,F10.5,I6,2F9.0,2F12.4
     1,F9.2,F13.3,F10.4,F10.3))
      PRINT 445,II
445   FORMAT ('1MAXIMUM PRODUCTION RANKING',10X,'BEST ',I3,' COMBINATION
     1S',/,T115,'*******',/,T115,'*      *')
      PRINT 440,(I,B(I,1,IDCON),B(I,2,IDCON),IR(I,3,IDCON),IR(I,IDCON),B(
     1I,4,IDCON),B(I,5,IDCON),B(I,6,IDCON),R(I,7,IDCON),B(I,8,IDCON),R((
     1,9,IDCON),B(I,10,IDCON),B((,11,IDCON),B(I,12,IDCON),I=1,II)
      PRINT 450,II
450   FORMAT ('1MAXIMUM PROFIT RANKING',10X,'BEST ',I3,' COMBINATIONS',/,T
     1124,'*******',/,T124,'*      *')
      PRINT 440,(I,O(I,1,IDCON),D(I,2,IDCON),ID(I,3,IDCON),ID(I,IDCON),O(
     1I,4,IDCON),O(I,5,IDCON),D(I,6,IDCON),D(I,7,IDCON),D(I,8,IDCON),O(I
     1,9,IDCON),O(I,10,IDCON),O(I,11,IDCON),O(I,12,IDCON),I=1,II)
455   CONTINUE
      RETURN
460   PRINT 465,NPR
465   FORMAT (1X,'PROGRAM ERROR --- PROBLEM NO. ',I3,' ABANDONED.')
      RETURN
      END
```

81

PROJECT NO. 1
JOB SPECIFICATIONS
LENGTH OF PART TO BE MACHINED   XL = 24.000IN.
DIAMETER TO START WITH   DIAS = 8.000IN.
DIAMETER TO FINISH AT   DIAF = 7.500IN.
RETURN SPEED OF THE CARRIAGE   FR = 5.IN./MIN.
TIME FOR FORWARD OR BACKWARD MOTION AT
    THE START OR END OF A CUT   T1 = 0.50000MIN.
TIME FOR LOADING AND
    UNLOADING THE WORKPIECE   TL = 5.0000MIN.
TOOL CHANGING TIME   TCT = 1.0000MIN.
TIME FOR CHANGING GEARS   TGC = 0.5000MIN.
COST FACTORS
LABOR RATE PLUS OVERHEAD   CO = 0.20000 $/MIN.
TOOL COST PER EDGE   CE = 0.50000 $/EDGE
MATERIAL COST   COSTMT = 300.0000 $/PIECE
SELLING PRICE   PRICE = 600.0000 $/PIECE
SPECIFIC HORSE POWER   SHP = 1.500H.P./CU.IN./MIN.
MACHINE TOOL TO BE USED
AVAILABLE HORSE POWER   HPM = 7.500H.P.
EFFICIENCY OF DRIVE ENERGY   EFF = 0.60000
SPINDLE SPEEDS AVAILABLE IN R.P.M. IN 50 STEPS
  20.   40.   60.   80. 100. 120. 160. 180. 200. 220. 240. 260. 280. 300. 320. 340. 360. 380. 400.
 420. 440. 460. 480. 500. 520. 540. 560. 580. 600. 620. 640. 660. 680. 700. 720. 740. 760. 780. 800.
 820. 840. 860. 880. 900. 920. 940. 960. 980. 1000.
FEEDS AVAILABLE IN IN./REV. IN 24 STEPS
0.0011 0.0015 0.0018 0.0024 0.0030 0.0036 0.0042 0.0046 0.0051 0.0056 0.0060 0.0068
0.0078 0.0084 0.0092 0.0094 0.0102 0.0112 0.0120 0.0128 0.0136 0.0147 0.0156 0.0168
SPEEDS TESTED IN F.P.M.
  42.   45.   49.   53.   57.   62.   67.   73.   79.   85.   92.   99. 107. 116. 126. 136. 147. 159. 172. 186.
 201. 218. 236. 255. 276. 298. 323. 349. 377. 408. 442. 478. 517. 559. 605. 654. 707. 765. 828. 895.
 969. 1048. 1133. 1226. 1326. 1434. 1551. 1678. 1815. 1963.

CONSTRAINTS ON CUTTING PARAMETERS:
DEPTH OF CUT (INCHES): 0.05000 < D < 0.25000
CUTTING SPEED (FPM): 80.0 < V < 200.0
FEED (I.P.R.): 0.00300 < F < 0.01680

TOOL LIFE DATA

OBSERVATION NUMBER	CUTTING SPEED (FPM)	FEED (IPR)	TOOL LIFE (MINUTES)
1	85.00	0.0078	26.2658

WITH ONLY ONE SET OF TOOL LIFE DATA AVAILABLE AND THE PARAMETERS ALPHA AND BETA OF THE TOOL LIFE EQUATION ASSUMED TO BE:
TOOL LIFE EXPONENT: ALPHA = 0.4000
FEED EXPONENT: BETA = 0.4000
THE TOOL LIFE CONSTANT, C, IS ESTIMATED TO BE:
    C = 45.083
ON THE ASSUMPTION THAT THIS TOOL LIFE EQUATION IS CORRECT, THE FOLLOWING TABLES GIVE THE BEST CONDITIONS FOR "MINIMUM COST"

```
************************
* 1. FINISHING CONDITIONS *
* (FEED UNDER 0.0080 IPR) *
************************
```

MINIMUM COST RANKING   BEST 20 COMBINATIONS

RANK	CUTTING SPEED INTENDED (FPM)	FEED (IPR)	DEPTH OF CUT (IN)	NO. OF PASS REQ.	RPM TO START WITH (RPM)	RPM TO FINISH WITH (RPM)	TOOL LIFE (MIN)	REMOVAL RATE IN CUTTING (CU IN/MIN)	HORSE POWER REQUIRED (HP)	TIME REQUIRED PER PIECE (MIN)	COST PER PIECE ($)	PROD. RATE (PC/HR)	PROFIT RATE ($/MIN)
1	126.	0.0078	0.25000	1	60.	60.	9.8670	2.9425	7.36	68.242	16.245	0.8792	4.158
2	116.	0.0078	0.25000	1	56.	56.	12.0071	2.7203	6.80	71.851	16.679	0.8351	3.943
3	107.	0.0078	0.25000	1	51.	51.	14.6113	2.5149	6.29	75.866	17.225	0.7909	3.727
4	136.	0.0068	0.25000	1	65.	65.	9.3008	2.7748	6.94	71.988	17.319	0.8335	3.927
5	126.	0.0068	0.25000	1	60.	60.	11.3181	2.5653	6.41	75.778	17.753	0.7918	3.725
6	99.	0.0078	0.25000	1	47.	47.	17.7804	2.3250	5.81	80.308	17.885	0.7471	3.513
7	159.	0.0060	0.25000	1	76.	76.	7.1182	2.8647	7.16	71.835	18.065	0.8352	3.925
8	116.	0.0068	0.25000	1	56.	56.	13.7729	2.3715	5.93	80.003	18.309	0.7500	3.521
9	147.	0.0060	0.25000	1	70.	70.	8.6621	2.6483	6.62	75.313	18.349	0.7967	3.740
10	172.	0.0056	0.25000	1	82.	82.	6.2674	2.8921	7.23	72.260	18.612	0.8303	3.894
11	92.	0.0078	0.25000	1	44.	44.	21.6369	2.1494	5.37	85.201	18.661	0.7042	3.302
12	136.	0.0060	0.25000	1	65.	65.	10.5409	2.4484	6.12	79.234	18.768	0.7573	3.549
13	159.	0.0056	0.25000	1	76.	76.	7.6267	2.6737	6.68	75.595	18.817	0.7937	3.720
14	107.	0.0068	0.25000	1	51.	51.	16.7601	2.1924	5.48	84.684	18.989	0.7085	3.318
15	147.	0.0056	0.25000	1	70.	70.	9.2809	2.4718	6.18	79.380	19.163	0.7559	3.538
16	126.	0.0060	0.25000	1	60.	60.	12.8271	2.2635	5.66	83.616	19.320	0.7176	3.357
17	186.	0.0051	0.25000	1	89.	89.	5.6552	2.8490	7.12	74.089	19.497	0.8098	3.786
18	85.	0.0078	0.25000	1	41.	41.	26.3497	1.9871	4.97	90.571	19.555	0.6625	3.096
19	172.	0.0051	0.25000	1	82.	82.	6.8818	2.6339	6.58	77.372	19.634	0.7755	3.624
20	136.	0.0056	0.25000	1	65.	65.	11.2938	2.2851	5.71	83.633	19.648	0.7174	3.352

****

IT IS NECESSARY TO COLLECT ADDITIONAL TOOL LIFE INFORMATION ON THIS OPERATION
BEFORE RELIABLE TOOL LIFE DATA CAN BE OBTAINED FOR FURTHER OPTIMIZATION

****

TRY ONE OF THE RECOMMENDED CUTTING CONDITIONS AND SUPPLY THE RESULTING TOOL LIFE DATA TO THIS PROGRAM FOR PROCESSING
- A NEW TOOL LIFE EQUATION WILL BE CALCULATED AND A NEW SET OF CUTTING CONDITIONS WILL BE GIVEN -

****

BASED ON THE 4 OBSERVATIONS AVAILABLE, THE PARAMETERS OF THE TOOL LIFE EQUATION ARE:
  TOOL LIFE EXPONENT: ALPHA = 0.3000
  FEED EXPONENT: BETA = 0.5300
  TOOL LIFE CONSTANT: C = 17.300
WITH THE NEW TOOL LIFE EQUATION, THE PERCENTAGE CHANGES IN THE PARAMETERS OF THE TOOL LIFE EQUATION ARE:
  CHANGE IN "ALPHA" = 0.000% (MAXIMUM VARIATION ALLOWED IS 5.00%)
  CHANGE IN "BETA" = 0.000% (MAXIMUM VARIATION ALLOWED IS 5.00%)
  CHANGE IN "C" = 0.001% (MAXIMUM VARIATION ALLOWED IS 5.00%)
****

  THE PERCENTAGE CHANGES ON THE PARAMETERS OF THE TOOL LIFE EQUATION ARE BELOW THE SPECIFIED
  MAXIMUM LIMITS. THEREFORE THIS TOOL LIFE EQUATION CAN BE ACCEPTED AS CORRECT.
  THE FOLLOWING TABLES GIVE THE BEST CONDITIONS FOR MINIMUM COST, MAXIMUM PRODUCTION RATE, AND MAXIMUM PROFIT RATE.
****

```
*********************************
*  I. FINISHING CONDITIONS      *
*     (FEED UNDER 0.0080 IPR)   *
*********************************
```

MINIMUM COST RANKING   BEST 20 COMBINATIONS

RANK	CUTTING SPEED INTENDED (FPM)	FEED (IPR)	DEPTH OF CUT (IN)	NO. OF PASS REQ.	RPM TO START WITH (RPM)	RPM TO FINISH WITH (RPM)	TOOL LIFE (MIN)	REMOVAL RATE IN CUTTING (CU IN/MIN)	HORSE POWER REQUIRED (HP)	TIME REQUIRED PER PIECE (MIN)	COST PER PIECE ($)	PROD. RATE (PC/HR)	PROFIT RATE ($/MIN)
1	116.	0.0078	0.25000	1	56.	56.	9.2496	2.7203	6.80	73.227	17.642	0.8194	3.856
2	126.	0.0078	0.25000	1	60.	60.	7.1195	2.9425	7.36	70.246	17.648	0.8541	4.019
3	107.	0.0078	0.25000	1	51.	51.	12.0169	2.5149	6.29	76.752	17.845	0.7817	3.676
4	99.	0.0078	0.25000	1	47.	47.	15.6121	2.3250	5.81	80.815	18.240	0.7424	3.486
5	126.	0.0068	0.25000	1	60.	60.	9.0723	2.5653	6.41	77.064	18.652	0.7786	3.651
6	136.	0.0068	0.25000	1	65.	65.	6.9831	2.7748	6.94	73.927	18.677	0.8116	3.805
7	92.	0.0078	0.25000	1	44.	44.	20.2829	2.1494	5.37	85.417	18.813	0.7024	3.292
8	116.	0.0068	0.25000	1	56.	56.	11.7866	2.2829	5.93	80.781	18.854	0.7428	3.480
9	107.	0.0068	0.25000	1	51.	51.	15.3129	2.1924	5.48	85.072	19.260	0.7053	3.300
10	85.	0.0078	0.25000	1	41.	41.	26.3511	1.9871	4.97	90.569	19.554	0.6625	3.096
11	136.	0.0060	0.25000	1	65.	65.	8.7113	2.4484	6.12	80.461	19.627	0.7457	3.485
12	147.	0.0060	0.25000	1	70.	70.	6.7052	2.6483	6.62	77.232	19.692	0.7769	3.629
13	126.	0.0060	0.25000	1	60.	60.	11.3175	2.2635	5.66	84.309	19.805	0.7117	3.323
14	99.	0.0068	0.25000	1	47.	47.	19.8943	2.0269	5.07	89.938	19.857	0.6671	3.115
15	159.	0.0060	0.25000	1	76.	76.	5.1611	2.8647	7.16	74.639	20.028	0.8039	3.751
16	147.	0.0060	0.25000	1	70.	70.	7.5744	2.4718	6.18	80.861	20.199	0.7420	3.460
17	116.	0.0056	0.25000	1	56.	56.	14.7035	2.0925	5.23	88.765	20.204	0.6759	3.152
18	136.	0.0056	0.25000	1	65.	65.	9.8405	2.2851	5.71	84.496	20.252	0.7101	3.311
19	159.	0.0056	0.25000	1	76.	76.	5.8301	2.6737	6.68	77.874	20.412	0.7705	3.590
20	126.	0.0056	0.25000	1	60.	60.	12.7846	2.1126	5.28	88.764	20.545	0.6759	3.148

DISTRIBUTION

Copies

A. Department of Defense

Defense Documentation Center
ATTN: TIPDR
Cameron Station
Alexandria, Virginia 22314   12

B. Department of the Army

Commanding General
U. S. Army Materiel Command
ATTN: AMCPP-PI
Washington, D. C. 20315   1

Commanding General
U. S. Army Tank-Automotive Command
ATTN: AMSTA-RCM.1, Mr. Don Phelps   1
       AMSTA-RCM.1, Mr. Edward Mortiz   1
Warren, Michigan 48090

Commanding General
U. S. Army Weapons Command
ATTN: AMSWE-PPR, Mr. F. X. Walter   1
       AMSWE-PPR, Mr. G. Hall   1
Rock Island, Illinois 61201

Commanding General
U. S. Army Missile Command
ATTN: AMSMI-IEL, Mr. L. Wheat
Redstone Arsenal, Alabama 35809   2

Commanding General
U. S. Army Munitions Command
ATTN: AMSMU-MT-MT, Mr. Julius Ziegler
Dover, New Jersey 07801   2

Commanding General
U. S. Army Electronics Command
ATTN: AMSEL-PP/P/IED.2, Mr. Wes Karg
225 South 18th Street
Philadelphia, Pennsylvania 19103   1

DISTRIBUTION

Copies

Commanding General
U. S. Army Electronics Command
ATTN: AMSEL-RD-GS, Mr. Stanley Mader
Fort Monmouth, New Jersey 07703   1

Commanding General
U. S. Army Aviation Materiel Command
ATTN: AMSAV-PRI, Mr. John Thorp   1
      AMSAV-PPL, Mr. F. Matthews   1
P. O. Box 209, Main Office
St. Louis, Missouri 63166

Commanding General
U. S. Army Mobility Equipment Command
ATTN: AMSME-PLA, Mr. J. J. Murphy   1
      AMSME-PLA, Mr. M. W. Schriet   1
4300 Goodfellow Blvd.
St. Louis, Missouri 63120

Commanding Officer
U. S. Army Materials and Mechanics Research Center
ATTN: AMXMR-TX, Mr. Arthur Jones   1
      AMXMR-TX, Mr. P. A. Carbonaro   1
Watertown, Massachusetts 02172

Commanding Officer
U. S. Army Aeronautical Depot Maintenance Center
ATTN: SAVAE-EFT, Mr. J. A. Dugan
Corpus Christi, Texas 78419   1

Project Manager, General Purpose Vehicles
Michigan Army Missile Plant
ATTN: AMCPM-GPV-QV, Mr. E. A. Cowgill
Warren, Michigan 48090   1

Commanding Officer
Watervliet Arsenal
ATTN: SWEWV-PPP-WP, Mr. L. Slawsky   1
      SWEWV-RDT, Library   1
Watervliet, New York 12189

Commanding Officer
Rock Island Arsenal
ATTN: SWERI-PPE-5311, Mr. J. A. Fox
Rock Island, Illinois 61201   1

86

DISTRIBUTION

Copies

Director
U. S. Army Production Equipment Agency
ATTN: AMXPE-MT, Mr. H. Holmes
Rock Island Arsenal
Rock Island, Illinois 61201   2

Commanding Officer
Savanna Army Depot
ATTN: AMXSV-EN
AMC Ammunition Center
Savanna, Illinois 61074   1

Commanding Officer
Fort Detrick
ATTN: SMUFD, Mr. H. H. Meier
Fredrick, Maryland 21701   1

AD   Accession No.
U.S. Army Weapons Command, Research, Dev. and Eng.
Directorate, Rock Island, Illinois 61201

COMPUTER OPTIMIZATION OF MACHINING CONDITIONS FOR
SHOP PRODUCTION, by Inyong Ham, Pennsylvania State
University, University Park, Pennsylvania

Report SWERR-TR-72-73, Oct 72, 87 p. incl. illus.
tables, (Contract OAAFO1-70-C-1D69, AMS Code
4932.06.6779) Unclassified report.

The work described in this report was initiated by
personnel of the Research Directorate, Weapons
Laboratory, U.S. Army Weapons Command, for applica-
tion of computer optimization techniques to the
study of machining parameters.

Two computer methods for industrial optimization of
machining conditions are described and demonstrated.
The PERFORMANCE INDEX METHOD (PIM) requires only

(Cont.) over

UNCLASSIFIED

1. Machining
2. Optimization
3. Computerization
4. Response surface
5. Regression analysis

DISTRIBUTION
Copies obtainable
from DDC

shop data for machining time, number of pieces produced, and number of tool changes. The PRODUCTION OPTIMIZATION METHOD (POM) requires tool life, time, and cost data. Both are designed to refine the initial data input with shop test data obtained during normal production, as related to one or more of three production objectives: minimum unit cost, maximum production rate and maximum profit rate. The computer programs are constructed for use by shop personnel with little knowledge of mathematics or computers.

Both methods are rapid and economical, and the programs can be processed by either in-plant or remote computer facilities. The user is given all information needed to install the programs and adapt them to his purposes.